Effective Video Streaming and Sharing in the Cloud

Kiran Mysa[1] & U.Mahender[2]

1. M-Tech CSE ,TKR College of Engineering and technology ,Meerpet. ,Hyderabad . Telangana.

2. Assistant Professor, TKR College of Engineering and technology ,Meerpet. , Hyderabad. Telangana.

[1]kiranmysaakil@gmail.com

Abstract—
This paper encouraged by recognition of the need of video traffic over mobile networks have been souring, the wireless link capability cannot keep up with the video traffic demand. Delay between the traffic demanding and the link capability, along with time-varying link environment, results in poor quality of video streaming service over mobile networks such as more buffering time and alternating disruptions. Leveraging the cloud computing technology, we suggest a new mobile video streaming framework, dubbed AMES-Cloud, which has two main parts: AMoV Adaptive mobile video streaming and efficient social video sharing. AMoV and ESoV assemble a private agent to provide video streaming services efficiently for each mobile user. For a given user, AMoV lets her private agent adaptively alter her streaming flow with a scalable video coding procedure 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 contented in advance. We implement a sample of the AMES-Cloud framework to make obvious its performance. It is shown that the private agents in the cloud can be effectively providing the video streaming.

KEYWORDS—
Effective video streaming; Mobile networks; AMES; sharing videos; cloud framework; Video quality

I.INTRODUCTION

Now a day, increasingly more traffic is accounted by the effective video streaming and downloading. In particular, video services over wireless mobile networks have become established over the past few Years. While demanding the video streaming is not so challenging in wired networks, mobile networks have been distress from video traffic transmissions over Insufficient bandwidth in wireless network links. Although network operators' desperate efforts to enhance the wireless link bandwidth (3G and LTE), soaring video traffic demands from mobile users are rapidly overpowering the wireless link capacity.

While receiving video streaming traffic via 3G/4G mobile networks, mobile users frequently suffer from long buffering time and intermittent disruptions due to the incomplete bandwidth and link condition instability caused by multi-path fading and user mobility. Thus, it is crucial to get better the service quality of mobile video streaming while using the networking and computing resources effectively. Recently there have been many studies on how to progress the service quality of mobile video streaming on following two aspects:

1.1 Scalability

Effective Mobile video streaming services should support a wide field of mobile devices; they have different video resolutions, different computing powers, various wireless links (like 3G and LTE) and so on. And Also, the available link capability of a mobile device may vary over time and space depending on its signals, other users traffic in the same cell, and link condition difference. Storing multiple versions 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 defines a base layer (BL) with multiple enhance layers (ELs). These substreams can be encoded by exploit three scalability features: spatial scalability by layering image resolution ,temporal scalability by layering the frame rate, and quality scalability by layering the image compression. By the SVC, a video can be played at the lowest quality if only the BL is delivered. However, even if the more ELs can be delivered, the better quality of the video stream is delivered.

1.2 Adaptability

In the traditional video streaming techniques designed by in view of relatively stable traffic links between servers and users, perform poorly in mobile environments. Thus the changeable wireless link status should be correctly dealt with to provide 'tolerable" video streaming services. To concentrate on this issue, we have to adjust the video bit rate adapting to the currently time-varying obtainable link bandwidth of each mobile user. Such adaptive streaming methods can effectively reduce packet losses and bandwidth wastage. Scalable video coding and adaptive streaming methods can be together combined to accomplish effectively the best achievable quality of video streaming services. That is, we can animatedly adjust the number of SVC layers depending on the current link status.

However most of the proposals in search of to jointly utilize the effective video scalability and adaptability rely on the active video control on the server side. That is, every mobile user needs to independently report the transmission status(like packet loss, delay and signal quality) sometimes to the server, which predicts the available bandwidth for each user. Thus the problem is that the server should take over the considerable processing overhead, as the number of users are increases.

II. RELATED WORK

2.1. Adaptive Video Streaming Methods

For this adaptive streaming, the video traffic rate is adjusted on the fly so that a user can experience the maximum possible video quality based on user link's time-changeable bandwidth capacity. There are mainly two types of adaptive streaming techniques, depending upon the whether the adaptively is controlled by the client or the server. The Microsoft's Smooth Streaming is a live adaptive streaming service which can control among different bit rate segments fixed with configurable bit rates and video resolutions at servers, while clients energetically request videos based on local monitoring of link quality. Adobe and Apple also developed client-side HTTP adaptive live streaming techniques operating in the similar manner. There are also some similar adaptive video streaming services where servers controls the adaptive transmission of video segments, for example, the Quavlive Adaptive Streaming. However, most of these solutions maintain multiple copies of the video contented with different bit rates, which brings enormous burden of storage on the server.

By considering the rate adaptation controlling techniques, TCP-friendly rate control methods for streaming video services over mobile networks are proposed, where TCP throughput of a flow is predicted as a function of packet loss rate , round trip time, and packet size. Considering the estimated output, the bit rate of the streaming traffic can be adjusted. Rate adaptation methods for informal 3G video streaming is introduced by. Then, a few cross-layer adaptation methods are discussed, which can acquire more accurate data of link quality so that the rate adaptation can be more accurately made. However, the servers have to always control and thus suffer from more workload.

Recently the Scalable Video Coding(SVC) technique has gained a momentum. An adaptive video streaming system based on SVC is developed in, which studies the real time SVC decoding and encoding at PC servers. The work in proposes a quality-oriented scalable video delivery using SVC, but it is only tested in a replicated LTE Network. Regarding the encoding performance of SVC, Cloud Stream mainly introduces to deliver high-quality streaming videos through a cloud-based SVC proxy, which find out that the cloud computing can significantly progress the performance of SVC coding.

2.2. Mobile Cloud Computing methods

These cloud computing methods has been well situated to provide video streaming services, especially in the wired Internet because of its scalability and capability. For example, the quality-guaranteed bandwidth auto-scaling for VoD streaming based on the cloud computing is proposed , and the CALMS framework is a cloud-assisted live media video streaming service for globally distributed users. However, extending the cloud computing services based on mobile environments requires more factors to consider: wireless link dynamics, user mobility, the restricted capability of mobile devices. More recently, new designs for users on top of mobile cloud computing environments are introduced , which virtualizes private agents that are in charge of satisfy in the requirements of individual users such as Cloudlets Stratus. Thus, we are motivated to design the AMES-Cloud framework by using virtual a gents in the cloud to provide effective video streaming services.

III.CLOUD FRAMEWORK FOR AMES

In this category we explain the AMES-Cloud framework which includes the Adaptive Mobile Video streaming(AMoV) and the Efficient Social Video sharing ESoV). From the whole video storing and video streaming system in the cloud is called as Video Cloud (VC). In this VC, there is a large-scale video base (VB), which is capable for storing the most of the popular video clips for the video service providers (VSPs). A temporal video base (tempVB) is used to cache new members for the popular videos, while tempVB counts the access frequency of each and every video. The VC keep maintains 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 first. By this two tier storage, the AMES-Cloud can keep serving most of popular videos eternally .Note that management work will be controlled by the controller in the VC.

Specially for each mobile user, a sub-video cloud (subVC) is created automatically if there is any video streaming demand from the user. The sub-VC has a sub video base (subVB), which stores the recently fetched video streaming segments. Note that the video deliveries among the subVCs and the VC in most cases are in fact not “copy”, but just “link” operations on the same file forever within the cloud data center. There is also encoding function in subVC. During live video streaming, mobile users will always report link conditions to their corresponding subVCs, and then the each subVCs offer effective video streams. Note that each mobile device also has a temporary caching storage, which is called local video base (localVB), and is used for buffering and also for perfecting.

Note that as the cloud services may across different places, or even locations, so in this case of a video delivery and pre-fetching between various 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. A comparison of the traditional video streaming, the scalable video streaming and the video streaming in the AMES-Cloud framework.

IV. ADAPTIVE MOBILE VIDEO STREAMING: AMOV

4.1. SVC

In traditional video streams with fixed bit rates are cannot adapt to the fluctuation of the link quality. For this reason a particular bit rate maintained , if the sustainable link bandwidth varies much and the video streaming can be frequently terminated due to the packet loss. In the SVC, a combination of the three lowest scalability is called the Base Layer (BL) while the improved combinations are called Enhancement Layers (ELs). To this regard, if BL is definite to be delivered, while more ELs can be also obtained when the link can give, a better video quality can be expected.

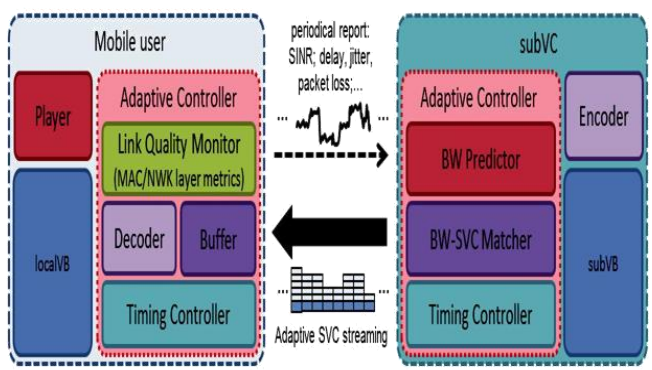

Fig 1.Client and sub VC structure

By using these SVC encoding techniques, the server doesn’t need to concern the client side and the link quality. Even some packets are lost, the client still can decode the effective video and display.

We are designing the mobile client and the subVC with the structure. The link quality monitor at mobile Client keeps tracking on metrics including video signal strength, packet round-trip-time (RTT), jitter and packet loss with a certain cycle. And the client will periodically reporting to the subVC. Hereby we define a cycle period for the reporting as the "time window", represented by T_{win}, Note that the video is also split by temporal segmentation.

Once again the subVC gets the information of the link quality, it will perform a computation and predict the potential bandwidth in the next time window. Note that we are using "predicted bandwidth" and "predicted goodput" interchangeably in following parts. Suppose let us consider sequence number of current time window is *i*, the predicted bandwidth can be estimated and where, i+i = 1 indicating the importance of each factor, *p* is for packet loss rate, *RTT* is for RTT, *SINR* is for the signal for interference and noise ratio, and *f*(), *g*(), *h*() are three functions reflecting the value change of each factor compared with that of last time window of bandwidth..

Actually in this paper we develop a measurement based prediction , that is we directly use $BW_i^{practical}$ of last time window as the $BW_{i+1}^{estimate}$ of next time window, which is proved with already high accuracy.

4.2 Matching between Bandwidth Prediction and SVC Segments

After obtaining the expected bandwidth, or say goodput, of next time window, subVC will match and decide how many effective video segments of BL and ELs can be transmitted roughly. We hereby define the term "resolution" to represent the level of temporal segmentation and the number of ELs. If T_{win} is small and there are more ELs, we can say that the SVC-based video source is with a higher resolution. We illustrate two cases of low resolution and a relatively high resolution for corresponding between the SVC segments and the predicted goodput. Matching between expected bandwidth and SVC-segments with various resolutions two ELs and a larger T_{win} can be hardly fit to the signal fluctuation, and thus there are some bandwidth wastage or packets lost. In contrast a higher resolution with more ELs and a smaller T_{win} can always fit the fluctuation of the bandwidth. However a higher resolution also includes more encoding workload to the servers.

Assume there are totally *j* ELs, and the bit rate of the *j*th EL is denoted as $R_{EL}j$ while the bit rate of the BL is R_{BL}.Let BL_i indicate the SVC segment of BL with sequential sequence *i*, and let EL^j_i specify the SVC segment of the *j*th EL with temporal sequence *i*. So the algorithm of matching between predicts bandwidth and SVC segments are shown in Algorithm.

V. FFICIENT SOCIAL VIDEO SHARING: ESOV

In these SNSs, users activate to known friends, famous people, and particular interested information publishers as well; also there are different types of social activities among users in SNSs, such as direct message and public posting. For spreading videos in SNSs, one can post a video in the public, and subscribers can quickly see it; one can also directly recommend a video to specified friend(s); further more one can periodically get observed by subscribed content publisher for new or popular videos.

Similar to researches in various strength levels for those social activities to represents 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 sub VCs can carry out effective background pre-fetching at subVB and even local VB. Because after a effective video sharing activity, there may be a certain delay that the recipient gets to know the sharing, and initiates to watch. Therefore the pre-fetching in prior will not effect the users at most cases. Instead, of a user can click to see without any loading delay as the beginning part or even the whole video is already pre-fetched at the local VB. The amount of pre-fetched segments is mainly determined by the strength of the various social activities. And the pre-fetching from VC to subVC only refers to the "linking" action, so there is only one file locating and linking operations with small delays; the pre-fetching from subVC to local VB also depends on the strength of the social activities, but will also consider the wireless link status.

We can categorize the social activities in current popular SNSs into three kinds, regarding the effect of the activities and the potential reacting priority from the different point of views of the recipient.

Activation: Like the popular RSS services, each user can activate to a particular video publisher or a special video collection service based on his/her own interests. This interest-driven connectivity between the various subscriber and the video publisher is considered as "median", because the subscriber may not always watch all subscribed videos.

Direct recommendation: In SNSs , an user directly recommend a video to particular person with a short message. The recipients of the message may watch it with high probability. This is measured as "strong".

Public Sharing: Each and every user in SNSs has a timeline-based of activity stream, which shows user 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 video with the "weak" connectivity among users, because not many people may watch the video that one has been seen without direct recommendation.

5.1 Pre-fetching Levels

Dissimilar strengths of the social activities indicate different levels of probability that a video will be quickly watched by the recipient. Correspondingly we also define three pre-fetching levels regarding the social activities of mobile users "Parts": Because of the videos that published by subscriptions may be watched by the users with a not high probability, we introduce to only push a part of BL and ELs segments, for example, the first 10% segments. "All": The video can be shared by the direct recommendations will be watched with a high probability, so we propose to pre-fetch the BL and all ELs, in order to let the recipient directly watch the required video with a good quality, without any buffering. "Little": The public sharing has a weak connectivity among various users, so the probability of that a user's friends (followers) watch the video that the user has watched or shared is low. We introduce to only prefetch the BL segment of the first time window in the beginning to those who have seen his/her activity in the video stream.

The pre-fetching happens among sub VBs and the VB, also more importantly, will be performed from the subVB to local VB of the mobile device depending on the link quality. If a mobile user is covered by Wi-Fi access, due to Wi-Fi's enable link and low price, subVC can push as much as possible in most cases. However even if it is with a 3G/4G connection, which charges a lot and suffers limited bandwidth, we introducing to downgrade the pre-fetching level to save energy and cost as listed, but users can still benefit from the pre-fetching effectively. Note that some energy prediction methods can be developed in order to actively decide whether current battery status is suitable for "parts" or "little". If a user, A, gets the direct recommendation of the video from another user, B, A's subVC will immediately pre-fetch the video either from B's subVB, or from the VB at the level of "all", if A is with Wi-Fi access.

However even 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 level of "parts". Note that the subscribed videos will be not pre-fetched when user A is at 3G/4G connection, as it is downgraded from "little" to none.

For a better extension of the pre-fetching strategy by social activities can be designed by self-updating mechanism from the user's hitting history in an evolutionary process. This learning-based pre-fetching 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

There are two parts, AMoV and EMoS, in AMES-Cloud framework that have tight connections and will together service the video streaming and video sharing, they both rely on the cloud computing platform and that are carried out by the private agencies of users. While pre-fetching in EMoS, the AMoV will still monitor and improve the transmission considering the link status; with a certain amount of pre-fetched video segments by EMoS, AMoV can offer better and efficient video quality.

With the efforts of AMoV and EMoS, we demonstrate the flow chart of how a video will be streamed. Note that in order to exchange the videos among the local VBs, sub VBs, tempVB and the VB, a video map is used to represent the required segments.

Once a mobile user starts to watch a required video by a link, the localVB will first be checked whether there are any pre-fetched segments of the video so that it can directly start. If there is none or just some parts of the client will report a corresponding VMap to its subVC. if the subVC has pre-fetched parts in subVB, the subVC will initiate the segment transmission. But if there is also none in the subVB, the tempVB and VB in the center VC will be checked. For this 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 a mobile user.

Also in the AMES-Cloud, if a video is shared by user 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 video shared at a much higher frequency (e.g., 100 times per day), it will be stored with a longer lifetime in the VB. In such a way, which is quite similar to the leveled CPU cache, the subVB and VB can always stored fresh and popular videos in order to increase the probability of the re-usage.

VII. IMPLEMENTATION AND EVALUATION

We can calculate the performance of the AMES Cloud framework by a replica implementation. We select the U-cloud server in the cloud computing examination offered by Korean Telecom, and utilize the virtual server with 6 virtual CPU cores (2.66GHz) and 32GB of memory, which is fast and sufficient for encoding 480P (480 by 720) video with H.264 SVC format in 30 fps at real time. In the cloud, we set up our server application based on the Java, including one main program handling to all tasks of the whole VC, while the program energetically initializes, maintains and terminates instance of different small Java application as the private agents for all active users. We can implement the mobile client at a mobile phone, Samsung Galaxy II, with android system version 4.1. The mobile data services are offered by LG U+ LTE network, while in some discovered area of the 3G network is usedWe can test in the downtown area, so the convenient bandwidth of the mobile link is not as high as we predictable, but this won't collision our experiment outcome.

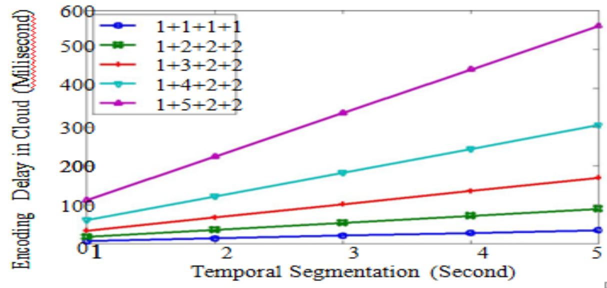

Fig.2 delay of difference in SVC resolution

This video test is the Tomb Raider 2012 Trailer in H.264 arrangement with 480P resolution downloaded from the YouTube. Its size is 13.849 Megabytes 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 software version 9.1. We just use defaulting settings to decode and encoding, and do the H-264 SVC encoding at the virtual server in the cloud.

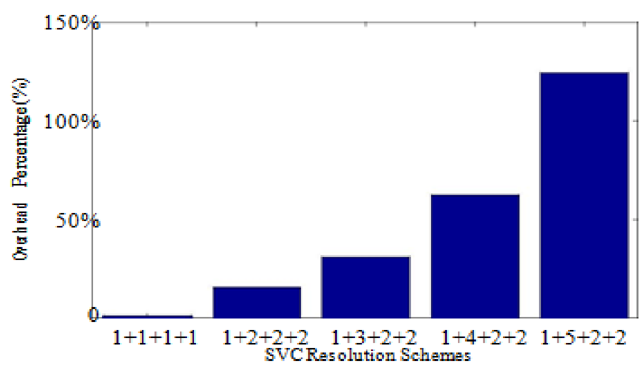

Fig.3 Overhead of Different SVC resolutions

We can divide the video into segments by 1 second to 5 seconds that is to vary T_{win} with values 1s, 2s, 3s, and 4s. By JSVM, besides the base layer, we further make five sequential 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 layers. Thus we define the best resolution configuration as "1+4+2+2". And we can also test different resolutions and configurations, including segment arrives, which is called as "click-to-play" delay. If the video has been cached in local VB, the video can be displayed almost immediately with ignorable delay. When we watch video which is fetched from the sub VC or the VC, normally it will take not more than 1 second to start. However if the user accesses to AMES-Cloud services via the cellular link, he will still suffer a bit longer delay due to the larger RTT of transmission via the cellular link.

VIII. CONCLUSION AND FUTURE SCOPE

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 for each mobile user to try to offer "non-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, subVBs and localVB of mobile users. We 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 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.

REFERENCES

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

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

[3] T.Taleb and K. Hashimoto, "MS2: A Novel Multi-Source Mobile-Streaming Architecture,"in *IEEE Transaction on Broadcasting*,

[4] X. Wang,S. Kim, T.Kwon, H. Kim, Y. Choi, "Unveiling the Bit Torrent Performance in Mobile WiMAX Networks," in *Passive and Active Measurement Conference*, 2011.

[5] 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.

[6] 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.

[7] 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.

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

[9] M. Wien, R. Cazoulat, A. Graffunder, A. Hutter, and P. Amon, "Real-Time System for Adaptive Video Streaming Based on SVC,"

[10] 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.

[11] 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.

[12] P. McDonagh, C.Vallati, A. Pande, and P. Mohapatra, "Quality-Oriented Scalable Video Delivery Using H. 264 SVC on An LTE Network," in *WPMC*, 2011.

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

[14] D. Niu, H. Xu, B. Li, and S.Zhao, "Quality-

Assured Cloud Bandwidth Auto-Scaling for Video-on-Demand Applications," in *IEEE INFOCOM*, 2012.

[15] Y.G. Wen, W.WZhang, K. Guan, D. Kilper, and H. Y. Luo, "Energy-Optimal Execution Policy for A Cloud-Assisted Mobile Application Platform," Tech. Rep., September 2011

[16] W.W Zhang, Y.G. and D.P.Wu, "Energy-Efficient Scheduling Policy for Collaborative Execution in Mobile Cloud Computing," in

[17] W.W.Zhang, Y.G. Wen, Z.Z. Chen and A.Khisti, "QoE-Driven Cache Management for HTTP Adaptive Bit Rate Streaming over Wireless Networks," in *IEEE Transactions on Multimedia*, November 2012.

[18] J. Li, M. Qiu, Z. Ming, G. Quan, X. Qin, and Z. Gu, "OnlineOptimization for Scheduling Preemptable tasks on IaaS Cloud systems," in *Journal of Parallel and Distributed Computing(JPDC)*, vol.72, no.5, pp.666-677, 2012.

[19] P. Calyam,M. Sridharan, Y. Xu , K. Zhu , A. Berryman, R. Patali, and A. Venkataraman, "Enabling Performance Intelligence for Application Adaptation in the Future Internet," in *Journal of Communication and Networks*, vol. 13, no. 6, pp. 591–601, 2011.

[20] Z. Huang, C.Mei, L. E.Li, and T. Woo, "CloudStream : Delivering High-Quality Streaming Videos through A Cloud-based SVC Proxy," in *IEEE INFOCOM*, 2011.

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

[22] B. Aggarwal, N. Spring, and A. Schulman, "Stratus : Energy-Efficient Mobile Communication using Cloud. .