International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878
Volume 2 Issue6, June 2014

AMES-Cloud: A Framework of Adaptive Mobile Video Streaming and Efficient Social Video Sharing in the Clouds

Shobha. D Jalikoppa[1]

P G Student, Department of Computer Science and Engineering, Gogte Institute of Technology, Belgaum, Karnataka, 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., prefetching) based on the social network analysis.***

Keywords: *S*calable video coding, Adaptive video streaming, Mobile networks, social sharing, Cloud computing

1. Introduction

Cloud computing promises lower costs, rapid scaling, easier maintenance, and services that are available anywhere, anytime. A key challenge in moving to the cloud is to ensure and build confidence that user data is handled securely in the cloud. A recent Microsoft survey found that "...58% of the public and 86% of business leaders are excited about the possibilities of cloud computing. But, more than 90% of them are worried about security, availability, and privacy of their data as it rests in the cloud."

There are many issues between user data protection and rich computation in the cloud. User wants to maintain control of their data, but also want to benefit from rich services provided by application developers using that data. At present, there is little platform-level support and standardization for verifiable data protection in the cloud. On the other hand, user data protection while enabling rich computation is challenging. It requires specialized expertise and a lot of resources to build, which may not be readily available to most application developers. It is argued that it is highly valuable to build in data protection solutions at the platform layer: The platform can be a great place to achieve economy of scale for security, by amortizing the cost of maintaining expertise and building sophisticated security solutions across different applications and their developers.

2. Target Applications

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 [1].. 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[2] [3] [4]. Thus, it is crucial to improve the service quality of mobile video streaming while using the networking and computing resources efficiently [5] [6] [7] [8]. Recently there have been many studies on how to improve the service quality of mobile video streaming on two aspects:

2.1 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 users' 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 [9] [10] [11] defines a base layer (BL) with multiple enhance layers (ELs). These substreams 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.

2.2 Adaptability

Traditional video streaming techniques designed by considering relatively stable traffic links between servers and users perform poorly in mobile environments. 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 [9] [12].

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. 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. 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 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.

An adaptive video streaming and prefetching framework for mobile users with the above objectives in mind, dubbed AMES-Cloud is designed. 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). 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 prefetched.

3. Literature Survey

Streaming and Sharing of Videos in Mobile Network

3.1 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.

3.2 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.

3.2.1 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.

3.2.2 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.

3.2.3 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.

3.3. 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.

3.3.1 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 [4]. The majority of transport protocols perform over an RTP stack, which is implemented transport for video streaming.

3.3.2 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.

3.4. Video Streaming Techniques

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

3.4.1 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, play-back 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 play-back until the data comes.

3.4.2 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. HLS streaming provides the best user experience, but its benefits also include good IT practices and important business considerations:

a) The best user experience.
b) Reach more viewers.
c) Save on data transfer.
d) Secure video content.

Cloud Front uses Adobe Flash Media Server 4.5 to stream on-demand content with Adobe's Real-Time Messaging Protocol (RTMP). Cloud Front accepts RTMP requests over port 1935 and port 80. Cloud Front supports the following variants of the RTMP protocol:

a) RTMP—Adobe's Real-Time Message Protocol.
b) RTMPT—Adobe streaming tunneled over HTTP.
c) RTMPE—Adobe encrypted over HTTP.
d) RTMPTE—Adobe encrypted tunneled over HTTP.
To secure it, just use the RTMPE protocol instead of the regular RTMP.

3.4.3 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 realtime. 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.

3.4.4 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.

3.5. Metrics Affecting Streaming: Quality

3.5.1 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 transcoding component and the user.

3.5.2 Metrics Characterizing Video Content

The characteristics of the video content can affect the trans-coding speed which decides the streaming quality. We use two cost-effective metrics to describe the video content heterogeneity: the temporal motion metric (TM) and the spatial detail metric (SD). TM can be captured by the differences of pixels at the same spatial location of two pictures, while SD is computed from the differences of all spatially neighboring pixels within a picture. Generally speaking, frequent scenery changes (large TM) affect the computation demand on video encoding by reducing the dependencies among the temporal successive pictures and increasing the number of pictures being intra-coded in response to a scenery change. Pictures with large SD can also increase the encoding computation overhead because of a demand for greater information and video clips have been returned.

4. Implementation

4.1 Matching Algorithm between BW and Segments

i = 0
BW0 = RBL
Transmit BL0

Monitor $BW_0^{practical}$
repeat
Sleep for Twin
Obtain p_i, RTTi, SINRi etc., from client's report
Predict $BW_{i+1}^{estimate}$ (or $BW_{i+1}^{estimate} = BW_i^{practical}$)
k=0
$BW_{EL}=0$
repeat
k++
if k >= j break
$BW_{EL}=BW_{EL} + R_{EL}k$
until $BW_{EL} >= BW_{i+1}^{estimate} - R_{BL}$
Transmit BL_{i+1} and EL^1_{i+1}, EL^2_{i+1},....... El^{k-1}_{i+1}
Monitor $BW_{i+1}^{practical}$
i++
until All video segments are transmitted.

4.2 Module Description

1. Admin Module
2. User1 Module
3. User2 Module

1. **Admin Module:** In this module, Admin have three sub modules. They are,

- **Upload Video:** Here Admin can add a new video. It is used for user for viewing more collections.
- **User Details:** Admin can view the user those have registered in this site.
- **Rate videos:** This module for avoiding unexpected videos from users. After accept/reject videos then only user can/cannot view their own videos.

2. **User1 Module:** In this module, it contains the following sub modules and they are,

- **News Feed:** Here user of this social site can view status from his friends like messages or videos.
- **Search Friends:** Here they can search for a friends and send a request to them also can view their details.
- **Share Video:** They can share videos with his friends by adding new videos also they share their status by sending messages to friends.
- **Update Details:** In this Module, the user can update their own details.

3. **User2 Module:**

In this module, user can register their details like name, password, gender, age, and then. Here the user can make friends by accept friend request or send friend request. They can share their status by messages also share videos with friends and get comments from them.

5. Conclusions

Here proposed is 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. Also AMES-Cloud can further seek to provide "nonbuffering" experience of video streaming by background pushing functions among the VB, subVBs and localVB of mobile users. The AMES-Cloud by prototype implementation 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. The cost of encoding workload in the cloud while implementing the prototype are ignored. As one important future work, the SNS-based prefetching, and security issues in the AMES-Cloud can improved.

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. 209–224, 2011.
[3] 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.
[4] X. Wang, S. Kim, T. Kwon, H. Kim, Y. Choi, "Unveiling the BitTorrent Performance in Mobile WiMAX Networks," in Passive and Active Measurement Conference, 2011.
[5] 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, "Multimedia Layout 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," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no. 9, pp. 1227–1237, Sep. 2007.

Author Profile

Ms. Shobha Jalikoppa is a P G Student, Department of Computer Science and Engineering, Gogte Institute of Technology, Belgaum. Karnataka, India

Under the guidance of
Prof. Sharada Kulkarni has done M. Tech in CSE. Presently working as Assistant Professor in Department of

Computer Science and Engineering, Gogte Institute of Technology, Belgaum, Karnataka, India. LMISTE, LMCSI

Prof. Shubhada Kulkarni has done M. Tech in CSE. Presently working as Assistant Professor in Department of Computer Science and Engineering, Gogte Institute of Technology, Belgaum, Karnataka, India. LMISTE, LMCSI