© 2015 IJSRSET | Volume 1 | Issue 1 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

SECURE VIDEO STREAMING IN SOCIAL NETWORKS USING CLOUD COMPUTING

Dhanya R[1], Anantharaman. G. R[2]
Department of CSE, Adhiyamaan College of Engineering, Hosur, India

ABSTRACT

Cloud Computing could be a technology that uses the internet and to maintain data and applications in central remote servers. Video streaming is one in every of the rising techniques in cloud. Video streaming implies that divide the total videos into type of section therefore transmitted the section into consumer. The video traffic over mobile network is increasing tremendously but the wireless link capability cannot maintain with the traffic. This gap ends up in poor service quality of video streaming over mobile networks. cloud computing technology into mobile networks, a replacement framework is introduced referred to as Secured AMES-Cloud containing two parts: AMoV (Adaptive Mobile Video streaming) and ESoV (Efficient Social Video sharing) .For each user, AMoV constructs a private agent to regulate streaming flow supported link quality using scalable video coding technique. ESoV permit social network interactions among users and private agents prefetch user requested videos before. Here, security is provided to each user in order that their videos cannot be seen by others unless the user desires.

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

I. INTRODUCTION

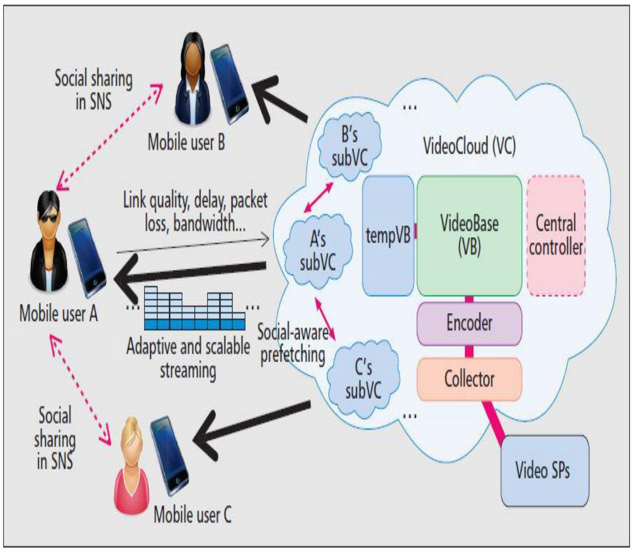
Cloud Computing permit consumers and businesses to use applications without installation and access their personal files at anytime and anywhere with internet access. For example of cloud computing is Yahoo email, Gmail, or Hotmail etc. All you would like is simple a web affiliation and you'll begin causing emails. The server and email management package is all on the cloud (internet) and is completely managed by the cloud service provider Yahoo, Google etc. Cloud computing is broken down into three segments: application, storage and connectivity. Every section serves totally different purpose and offers different merchandise for businesses and people round the world.

Today, video streaming is one among the foremost widespread services on the internet [1]. The increasing demand for video streaming has meant video constitutes an outsized portion of the overall information traffic on the Internet. Video streaming suggests that divide the full videos into variety of section so transmitted the section into consumer [1]. Whereas sending the videos in server, consumer side can automatically produce the buffer for storing the divide section. If one buffer is full, video will begin to play and automatically produce another buffer for storing remaining section.

II. METHODS AND MATERIAL

A. RELATED WORK

Over the past decade, additional more traffic is accounted by video streaming and downloading. Specifically, video streaming services over mobile networks become rife over the past few years [1]. Whereas the video streaming isn't thus difficult in wired networks, mobile networks are stricken by video traffic transmissions over scarce information measure of wireless links. Despite network operators' desperate efforts to boost the wireless link information measure (e.g., 3G and LTE), soaring video traffic demands from mobile users area unit quikly overwhelming the wireless link capacity. Whereas receiving video streaming traffic via 3G/4G mobile networks, mobile users typically suffer from long buffering time and intermittent disruptions attributable to the restricted information measure and link condition fluctuation caused by multi-path craze and user quality. Thus, it's crucial to enhance the service quality of mobile video streaming whereas using the networking and computing resources efficiently.



Scalability

Video Streaming service should be compatible with multiple mobile devices having numerous video resolutions, computing powers, wireless links then on. Capturing multiple bit rates of same video might increase the burden on servers in terms of storage and sharing. To resolve this issue, the Scalable Video Coding (SVC) technique has been introduced. Scalable Video Coding (SVC) is that the name for the Annex G extension of the H.264/MPEG-4 AVC video compression standard. SVC standardizes the encryption of a high-quality video bit stream that also contains one or additional set bitstreams. A set video bitstream comes by dropping packets from the larger video to scale back the information measure needed for the set bitstream. A set bitstream will represent a lower spatial resolution, or a lower temporal resolution, or a lower quality video signal (each separately or in combination) compared to the bitstream it is derived from.

The following modalities are possible:

- Temporal (frame rate) scalability: The motion compensation dependencies are structured so complete pictures (i.e. their associated packets) can be dropped from the bit stream.
- Spatial (picture size) scalability: Video is coded at multiple spatial resolutions. The data and decoded samples of lower resolutions can be used to predict data or samples of upper resolutions so as to scale back the bit rate to code the upper resolutions.
- SNR/Quality/Fidelity scalability: Video is coded at one spatial resolution however at completely different qualities. The data and decoded samples of lower qualities can be used to predict data or samples of higher qualities in order to reduce the bit rate to code the higher qualities.
- Combined scalability: A mixture of the three scalability modalities described above.

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. Cloud computing techniques are used to provide scalable resources to service providers to serve mobile users. Hence, clouds are used for large scale real time video services. Many Mobile cloud computing technologies have provided private agents for serving mobile users e.g., Cloudlet. This is because, in cloud multiple threads can be created dynamically based on user demands.

Social Network Services (SNS's) have occupied a major role recently. In SNS's user can share, comment, and post the videos among friends and groups. Users can follow their favourites depending on their interest in which their followers are likely to watch popular person posts. E.g., Twitter, Facebook.

Scalable Video Coding

Delivering video stream using different resolutions to satisfy different client needs/constraints.

Multi-Layer Coding (Multi-resolution)

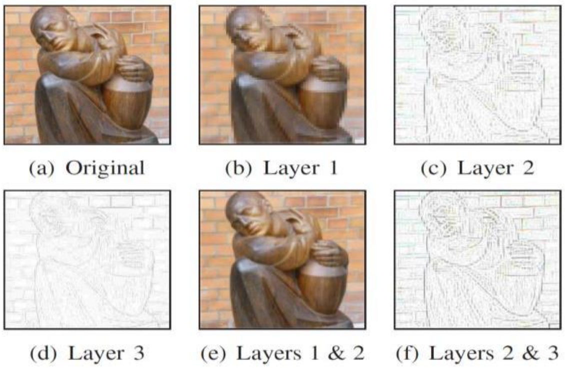
- Base layer
- Enhancement layers

Multiple Descriptions Coding (MDC)

- Multiple independent video substreams
- Receiving more substreams increases the video quality

Substream_1	Resolution_1

Substream_2	Resolution _2
Substream_N	Resolution _N


(a) Original (b) Layer 1 (c) Layer 2
(d) Layer 3 (e) Layers 1 & 2 (f) Layers 2 & 3

Algorithm:
Matching Algorithm between BW and Segments

$i = 0$

$BW_0 = R_{BL}$

Transmit BL_0

Monitor BW_0 *practical*

repeat

  Sleep for *Twin*

  Obtain *pi*, *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

B. ESOV: EFFICIENT SOCIAL VIDEO SHARING

A. Social Content Sharing

In SNSs, users subscribe to known friends, famous people, and specific interested content publishers as well; conjointly there square measure numerous style of social activities among users in SNSs, like as direct message and public posting. For spreading videos in SNSs, one will post a video within the public, and his/her subscribers will quickly see it; one can also directly suggest a video to specified friend(s); moreover one can periodically get detected by signed content publisher for brand new or standard videos. Similar to studies in, we have a tendency to outline totally different strength levels for those social activities to point the probability that the video shared by one user could also be watched by the receivers of the one's sharing activities, that is named a "hitting probability", so subVCs will do effective background prefetching at subVB and even localVB. Because after a video sharing activity, there could also be an exact delay that the recipient gets to understand the sharing, and initiates to look at. So the prefetching in previous won't impact the users at the most cases. Instead, a user will click to check with none buffering delay as the beginning part or even the whole video is already prefetched at the localVB. The amount of prefetched segments is mainly determined by the strength of the social activities. And the prefetching from VC to subVC only refers to the "linking" action, thus 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:** Just like the standard RSS services, a user will buy a specific video publisher or a special video collection service based on his/her interests. This interest-driven connectivity between the subscriber and therefore the video publisher is considered as "median", as a result of the subscriber might not perpetually watch all subscribed videos.

• **Direct recommendation:** In SNSs, user directly suggests a video to specific friend(s) with a brief message. The recipients of the message might watch it with terribly high probability. This is often thought about as "strong".

• **Public sharing:** Each user in SNSs encompasses a timeline-based of activity stream that shows his /her recent activities. The activity of a user look or sharing a video will be seen by his/her friends (or followers). We have a tendency to contemplate this public sharing with the “weak” property among users, as a result of not many of us might watch the video that one has seen while not direct recommendation.

B. Prefetching Levels

Different strengths of the social activities indicate totally different levels of probability that a video are going to be shortly watched by the recipient. Correspondingly we have a tendency to conjointly define three prefetching levels relating to the social activities of mobile users:

• **“Parts”:** Because the videos that published by **subscriptions** could also 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** are going to be watched with a high probability, thus we have to tendency to propose to prefetch the BL and everyone ELs, in order to let the recipient(s) directly watch the video with a honest quality, with none any buffering.

• **“Little”:** The **general public sharing** encompasses a weak property among users, that the probability that a user’s friends (followers) watch the video that the user has watched or shared is low. We propose to only prefetch the BL phase of the primary time window within the setting out to those that have seen his/her activity within the stream.

III. RESULTS AND DISCUSSION

We assess the performance of the AMES-Cloud framework by a prototype implementation. We decide the U-cloud server (premium) within the cloud computing service offered by Korean medium, and utilize the virtual server with six virtual CPU cores (2.66 GHz) and 32 GB memory, that is quick enough for secret writing 480P (480 by 720) video with H.264 SVC format in 30 fps at real time. Within the cloud, we have a tendency to deploy our server application supported Java, as well as one main program handling all tasks of the total VC, whereas the program dynamically initializes, maintains and terminates instances of another tiny Java application as private agents for all active users. We have a tendency to implement the mobile client at a mobile phone, Samsung Galaxy II, with android system version 4.0.

The mobile data service is obtainable by LG LTE network, whereas in some uncovered space the 3G network is employed. Note that we have a tendency to use “3G” to point the overall cellular network. We check within the downtown space, that the sensible information measure of the mobile link is not as high as we have a tendency to expected, however this won’t impact our experiment results. The check video is that the topographic point Raider 2012 Trailer in H.264 format with 480P resolution downloaded from YouTube. Its size is 13.849 Mbytes and with a period of 180 seconds. We have a tendency to 1st decipher 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 simply use default settings for the decoding and encoding, and do the H.264 SVC encoding at the virtual server within the cloud. We have a tendency to split the video into segments by 1 second to 5 seconds, that’s 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). Therefore we have a tendency to outline the simplest resolution configuration as “”. And we also test different resolution configurations, including “”, “”, “” and “”.

A. Adaptive Video Streaming Based on SVC

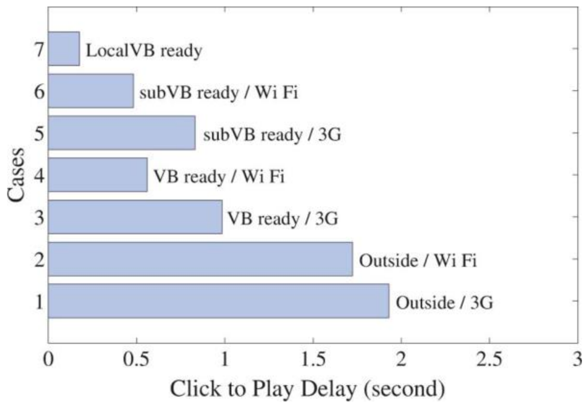
Firstly we have a tendency to examine whether or not there’s a deep relationship between the measured information measure of last time window and therefore the sensible information measure of next time window (good put by Kbps). We have a tendency to check the video streaming service via cellular link, and move the device around within the building to undertake to vary the signal quality. The collected the relative errors for the expected information measure to the sensible bandwidth for each time window, wherever 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 expected information measure is almost the sensible one with around 10% relative error, however giant values of have comparatively poor prediction accuracy, which reflects the similar results. Thus, we advise a brief of 2 or 3 seconds for accurate prediction in sensible style.

B. Video Streaming in SubVC and VC

We assess however H.264 SVC works in AMES-Cloud framework relating to the on top of mentioned SVC resolution configurations. Due to the robust procedure capability by the cloud computing, the encoding speed is quick. The best resolution configuration “” with 5 second temporal segmentation scheme requires about 560 ms for encoding. For shorter intervals of, the encoding delay is very small less than 50 ms.

C. Prefetching Delays

In ESoV, video segments are often perfected among VB, tempVB, and localVBs of the mobile users, supported their activities in SNSs. we tend to evaluate the desired delays for various levels of prefetching.



We tend to here use the normal resolution configuration of “ ” with 2 second temporal segmentation by default (the same in following tests). We tend to conjointly 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 which of “all” as all BL and ELs segments.

IV. CONCLUSION

In this paper, we mentioned our proposal of an adaptive mobile video streaming and sharing framework, referred to AMES-Cloud, that efficiently stores videos within the clouds (VC), and utilizes cloud computing to construct private agent (subVC) for every mobile user to undertaken to supply “non-terminating” video streaming adapting to the fluctuation of link quality supported on the Scalable Video Coding technique. Also AMES-Cloud will additional ask for to supply “non-buffering” experience of video streaming by background pushing functions among the VB, subVBs and localVB of mobile users. We tend to 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 are going to perform large-scale implementation and with serious thought on energy and price cost. In the future, we will also try to improve the SNS-based prefetching, and security problems within the AMES-Cloud.

V. REFERENCES

[1] Xiaofei Wang, Min Chen, Ted Taekyoung Kwon, Laurence T. Yang, and Victor C.M. Leung, “AMES-Cloud: A Framework Mobile Video Streaming and Efficient Social Video Sharing in the Clouds”, IEEE Transactions on cloud computing , vol:15 , no:4, year 2013.

[2] Sahel Mastoureshgh, a Thesis of “Measurement and Method for Receiver Buffer Sizing in Video Streaming”, in April 2012.

[3] Patrick Seeling and Martin Reisslein, “The Rate Variability-Distortion (VD) Curve of Encoded Video and Its Impact on Statistical Multiplexing”, IEEE Transactions on broadcasting, vol. 51, no. 4, December 2005.

[4] Gabriel-Miro Muntean, Gheorghita Ghinea and Timothy Noel Sheehan, “Region of Interest-Based Adaptive Multimedia Streaming Scheme”, IEEE Transactions on broadcasting, vol 54, no. 2, June 2008.

[5] K.Tappayuthpijarn, G. Liebl, T. Stockhammer, and E. Steinbach, “Adaptive Video Streaming over A Mobile Network with TCP-Friendly Rate Control”, in IWCMC, 2009.

[6] R.Abinaya, G.Ramachandran, “Efficient p2p Video Sharing Scheme in Online SocialNetwork”, **v**ol.3, Issue 5, pp. 23-27, 2014.

[7] M.Sona, D.Daniel, S.Vanitha, “A Survey on Efficient Video Sharing and Streaming in cloudEnvironment using vc”, Vol. 1, Issue 8, 2003.