Journal of Advances in Mathematics and Computer Science

34(2): 1-11, 2019; Article no.JAMCS.46761
ISSN: 2456-9968
(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)

A Secured Adaptive HTTP-Based Video Streaming System Using AES Algorithm

Aderonke J. Ikuomola[1*]

[1]Department of Mathematical Sciences, Ondo State University of Science and Technology, Okitipupa, Nigeria.

Author's contribution

The sole author designed, analysed, interpreted and prepared the manuscript.

Article Information

DOI: 10.9734/JAMCS/2019/v34i230213
Editor(s):
(1) Dr. Mohd Zuki Salleh, Professor, Universiti Malaysia Pahang, Malaysia.
(2) Dr. Octav Olteanu, Professor, Department of Mathematics-Informatics, University Politehnica of Bucharest, Romania.
(3) Dr. Kai-Long Hsiao, Associate Professor, Taiwan Shoufu University, Taiwan.
Reviewers:
(1) Mudasir Manzoor Kirmani, Sher-e-Kashmir University of Agricultural Sciences & Technology, India.
(2) Zlatin Zlatev, Trakia University, Bulgaria.
(3) Degan Zhang, Tianjin University of Technology, China.
Complete Peer review History: http://www.sdiarticle4.com/review-history/46761

Received: 02 April 2019
Accepted: 10 June 2019
Published: 26 October 2019

Original Research Article

Abstract

Streaming video is a process of transferring moving images or video over the internet in a compressed form to viewers so that it can be viewed in real time. Streaming video technology is becoming more powerful all the time and appreciable works have been done in this area. However, security and adaptation of video are still problems to be tackled in the area of video streaming. In this paper, a secured and adaptive HTTP video streaming system (SAHVISS) was designed. The system was implemented using PHP, HTML5, Angular JS, Java Script and SQL Server database. It was tested on different computers and mobile devices using various web browsers. SAHVISS consist of four components namely Video Source, Streaming Server, Distribution Server and Client. The Streaming Server is made up of two sub-components namely the media encoder and the streamer. The server requires a media encoder to converts raw (uncompressed) digital video to a compressed format while the streamer breaks the encoded media into segments and save them as file. The distribution server comprises of Content Delivery Network and the Web server which delivers media file and the index files to the client over Hyper Text Transfer Protocol (HTTP). The performance of SAHVISS shows that it provides an efficient technique for solving the issues of adaptation and privacy/security of streaming video content. It is a robust, secured and an adaptive http video streaming system.

**Corresponding author: E-mail: deronikng@yahoo.com;*

Keywords: Adaptive; content delivery; security; streaming video.

1 Introduction

Video has been a means of information exchange and entertainment between people in society for many years. Before now video was initially recorded and sent in analogue form and this was on for many years before the arrival of computers and digital integrated circuits which resulted into video digitization and this enabled a revolution in the compression and communication of video. As a result of the increase and high demand of the Internet and its users in the mid 1990's, the interest of people in video communication over the internet network also increases.

Digital video is a depiction of moving visual images in the form of encoded digital data. This is in contrast to analogue video, which represents moving visual images with analogue signals. With the increasing accessibility of technology by people every day, things are beginning to get digitalized: digital camera, digital cable, digital sound and digital video [1]. Due to the size of some large video files and the limited bandwidth, transmission of video and audio data via the Internet is only possible using streaming technology. Streaming video is a sequence of moving images, which are transferred in the compressed form and sent over the Internet to viewers so that they can display it on the screen as they arrive [2]. Video streaming is concerned with delivering video data from server to client over the network as fast and with as little loss as possible. If video data is received by an end user as it streams, then users do not have to wait to download a large set of file, before watching the video or listening to the audio [3].

There exist a very diverse range of different video communication and streaming applications, which have very different operating conditions or properties. For example, video communication application may be for point to point communication or for multicast or broadcast communication, and video may be pre-encoded (stored) or may be encoded in real-time (e.g. interactive videophone or video conferencing). The video channels for communication may also be static or dynamic, packet-switched or circuit switched, may support a constant or variable bit rate transmission, and may support some form of Quality of Service (QoS) or may only provide best effort support. The specific properties of a video communication application strongly influence the design of the system [4].

In recent years, video streaming services such as YouTube, Dailymotion and Veoh have become more and more popular. These services communicate using Hypertext Transfer Protocol (HTTP) which can easily be used to watch video sequence using a web browser, if Adobe Flash Player is installed on computer or web browser supports Hypertext Markup Language 5 (HTML5-the next generation of HTML). The HTTP uses Transmission Control Protocol (TCP) as its transport-layer protocol [1].

However, video streaming service should be provided in a way that it will suit the users, several challenges users have faced in the area of video streaming include among others; bandwidth and video quality, format conversion, user's authentication and security, live streaming of videos and platform access restrictions.

No doubt, appreciable works have been done in the area of video streaming. However, security and adaptive video is still a problem to tackle in the area of video streaming. Videos should able to change themselves in order to fix themselves better in some new environment in case of streaming globally [5]. This prompt the idea of designing a Secure and Adaptive HTTP Video Streaming System to foster systematic interaction between users with the purpose of enhancing the quality video and maximum security among users while streaming videos over the internet domain. This is done with the intent of establishing an indispensable internet domain that offers users on various platforms maximum protection and high-quality adaptive videos.

In this work, a Secured and Adaptive HTTP-Based Video Streaming System using AES Algorithm was developed. In the rest of this paper, section 2 presents the review of related works done on video streaming, Section 3 describes the technique employed in the design of a Secured and Adaptive HTTP-Based Video Streaming System. Section 4 and 5 describe the implementation and conclusion respectively.

2 Literature Review

Zhu X. and Girod B. [6], reviewed key problems and tentative solutions for video streaming over wireless networks, with an emphasis on network-adaptive rate control and resource allocation among multiple video streams. Cross-layer information exchange is required, so that video source rates can adapt to the time-varying wireless link capacities. To optimally allocate network resource among heterogeneous traffic types, each bearing a different performance metric (e.g., the completion time for file downloading versus video quality for streaming) is a major challenge. It is still unclear whether the stringent latency constraint (usually less than a second) for video streaming can be met when packets need to be delivered over multiple hops of time-varying wireless links in a mesh network.

Hiroki O. et al. [1] proposed a new transport-layer protocol for video streaming, called TCP Stream. TCP Stream performs window-based congestion control that combines two congestion controls: a loss-based congestion control that uses packet loss as an index of congestion and a delay-based congestion control that uses a network delay as an index of congestion. TCP Stream can utilize open bandwidth when a network is or not in a congestion state. It transmits data packets at an adjusted rate required for the video sequence, unlike TCP NewReno, and does not steal bandwidth from other network traffic [7]. In this work, an effective mechanism was not used because the experiment was not implemented and evaluated over the internet.

Bailey J. M. [8] developed an open source solution capable of transferring the live video with little overhead on the phone and/or server. Where users will have the ability to broadcast news and events live using only Android-enabled mobile devices and an internet connection via the cellular network or WiFi. Developers will have access to suggest changes to the source code, paving the roads for new innovative ideas based on the technology and personal users and enterprises will have complete control over where the video is transferred over the internet. In Android 3.0, Google introduces H.264 AVC codec, H.264 has the highest quality but consumes more uploading bandwidth as well as more phone and server power.

Sabu M. T. [9] did a reviewed on P2P Video Streaming**.** The main benefit of P2P streaming is that each peer contributes its own resources to the streaming session. Administration, maintenance, and responsibility for operations are hence dispersed among several users instead of focusing on few servers. Due to this, there is a rise in the quantity of resources in the network. The client-server design harshly restricts the number of concurrent users in video streaming due to the bandwidth bottleneck at the server side. Security has a significant impact on P2P based streaming applications. Media streaming is inherently more prone to attacks as it is very difficult to monitor the participating peers in the overlay.

Pradan N. L. [10] surveyed major approaches and mechanisms for Internet video streaming and presented an adaptive framework for video over wireless IP. In a multicast scenario, receivers may have different requirements and properties in terms of latency, visual quality, processing capabilities, power limitations (wireless vs. wired) and bandwidth limitations. The heterogeneous nature of receivers' requirements and properties make it difficult to design an efficient multicast mechanism. Compared with the wired links, wireless channels are typically noisier and have both small-scale (multipath) and large-scale (shadowing) fade, making the bit error rate (BER) very high. The resulting bit errors can have a devastating effect on video presentation quality.

Android-based application "Media Streaming" was created by Patil J. [11] for the World Wide Web users to stream their choice of videos, securely. The application is supported through user authentication before accessing the videos available on the Web store. The video streaming design using security uses minimal processing with little overhead while maintaining security. The Author's Infrastructure widens over a diverse computers windows operating system, an Android platform and various software packages. Few factors like bandwidth and video quality have not been taken into consideration during the development and performance testing of this application. As video streaming is managed via HTTP, the speed and efficiency also depend upon the network bandwidth. LIVE content streaming of videos was not addressed.

Junchen J. et al. [12] revisited classical networking problems with respect to resource sharing and adaptation. The authors work within the constraints that have spurred the growth of video traffic using HTTP, no modifications to end-host stacks, and imposing no modification to the network and CDN server infrastructure. Within this context, they provide a clearer understanding of problems that lead to inefficiency, unfairness and instability when multiple players compete for a bottleneck link. Building on these insights, they provide guidelines on designing better scheduling and bitrate selection techniques to overcome these problems.

Kim Y. et al. [13] proposed a network-adaptive HTTP video streaming system over Wi-Fi and 3G mobile networks. The streaming system adopts the version of HTTP live streaming from the Internet Engineering Task Force (IETF). In addition, the system consists of throughput estimation and adaptive video rate selection, which is enhanced to ensure the quality of service (QoS) as well as improve efficiency. HTTP live streaming is available with a general web server and is easy to implement, but managing streaming data is difficult because all the segmented streams and metadata must be stored in separate files. Therefore, the server must handle large-scale file management as well.

Oda H. et al. [14] proposed a new transport-layer protocol, called TCP Stream, that solves the problem of TCP in video streaming. TCP Stream performs an hybrid congestion control that combines the loss-based congestion control, which uses packet loss as an index of congestion, and the delay-based congestion control, which uses the delay as an index of congestion.

Krishnamoorthi V. [15] work is to establish a causal relationship between video quality and viewer behaviour, taking a step beyond purely correlational studies. In order to establish causality, the authors use Quasi-Experimental Designs, a novel technique adapted from the medical and social sciences. The authors study the impact of video stream quality on viewer behaviour in a scientific data-driven manner.

By inculcating cloud computing technology into mobile networks, a new framework is introduced by Manasa V. and Vikram M. [16] called Secured AMES-Cloud which contains two parts: Adaptive Mobile Video streaming (AMoV) and Efficient Social Video sharing (ESoV). For each user, AMov constructs a private agent to adjust streaming flow based on link quality using scalable video coding technique. ESoV allows social network interactions among users and private agents' prefetch user requested videos in advance. Here, security is provided to each user so that their videos cannot be seen by others unless the user requires and cannot be seen by cloud providers using Homomorphic and Incremental encryption [17].

Mobile users are affected by signal strength because of it low configurations. So, in order to tackle this traffic demands, [5] proposed an approach for sharing secured and efficient video streaming over the cloud. The Cloud Computing own and manages mobile users' data resulting in fast data access and storage, and also give faster access to videos.

Shuai Y. [18] addressed the minimum buffering size required in adaptive streaming, which provides the guidelines to determine a reasonable low latency for streaming systems. Then, the author tackles the fundamental challenge of achieving such low-latency streaming by developing a novel adaptation algorithm that stabilizes buffer dynamics despite a small buffer size. A novel adaptation architecture with low-delay feedback for the bitrate selection and optimizing the underlying transport layer to offer efficient real-time streaming was designed.

In the context of Content Delivery Network (CDN) nodes or proxies, [15] investigates the interaction between HAS clients and proxy caches. The author proposed and evaluate classes of content-aware and collaborative policies that take advantage of information that is already available or share information among elements in the delivery chain, where all involved parties can benefit.

3 Methodology

3.1 Architecture of a secured and adaptive HTTP video streaming web service

The architecture of a secured and adaptive HTTP Video streaming web domain (SAHVISS) is shown in Fig. 1. SAHVISS consist of four components namely Video Source, Streaming Server, Distribution Server and Client.

SAHVISS allows compressed video content to be transferred via internet in order to please the client (user) curiosity as well as expectations. Input can be live or from a prerecorded source. The streaming server is responsible for taking input streams of media which can either be live or prerecorded and then encode them digitally, encapsulating them in a format suitable for delivery, and preparing the encapsulated media for dissemination. The streamer breaks the encoded media into segments and saves them as files using media stream software. At the distribution, server is the Content Delivery Network (CDN) and the web server. The goal of the CDN is to deliver the video content to the client (end-users) with high availability and performance. CDNs deliver a large fraction of the Internet content today (live streaming media, on-demand streaming media, etc.). The clients can visit SAHVISS web pages on his device using the Hypertext Transfer Protocol (HTTP) and then makes a request to the CDN through the Domain Name System (DNS). The video delivered to the clients is viewed in real time using Adaptive HTML5 video player available on SAHVISS.

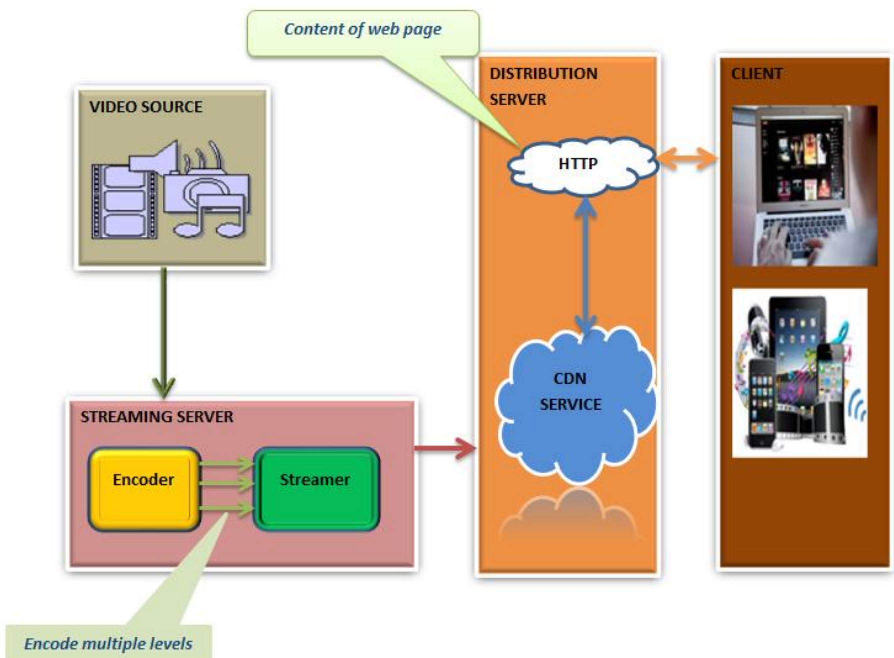

Fig. 1. Architectural design of a secured and adaptive HTTP video streaming web service

3.1.1 Component of the architectural design

(a) **Video Source:** Input can be live or from a prerecorded source. It is typically encoded into an MPEG-2 Transport Stream which is then broken into segments and saved as a series of one or more .ts media files.

(b) **Streaming Server:** is made up of two sub-components namely the media encoder and the streamer. The server requires a media encoder to convert raw (uncompressed) digital video to a compressed format while the streamer breaks the encoded media into segments and save them as files.

(i) **Encoder:** The media encoder takes a real-time signal from an audio-video device, encodes the media (that converts raw digital video to a compressed format), and encapsulates it for delivery. The compressed video format usually conforms to a standard video compression specification. The compression is typically loosening, meaning that the compressed video lacks some information present in the original video.

(ii) **Streamer:** is software that reads the Transport Stream sent from the media encoder and divides it into a series of small media files of equal duration. Although, each segment is in a separate file each video files are made from a continuous stream which can be reconstructed without any disruption.

(c) **Distribution Server:** comprises of Content Delivery Network and the Web server. It delivers media file and the index files to the client over HTTP.

(i) **Content Delivery Network Service (**CDNS): is to serve content to the clients with high availability and high performance. It operates as an ASP on the Internet (also known as on-demand software or software as a service (SaaS)). This offers access to media streaming to SAHVISS subscribers/Clients. Here content (potentially multiple copies) may exist on several servers. When a client makes a request to a CDN hostname, Domain Name System (DNS) will resolve to an optimized server and the server will handle the request.

(ii) **Web Server:** delivers SAHVISS web pages in response to the requests sent by clients using the Hypertext Transfer Protocol (HTTP). These Web pages are simply HTML documents along with additional content such as images, style sheets and scripts

(d) **Client:** connects to SAHVISS web server through http://sams.net and choose videos of their choice. The HTML5 (Adaptive video player) specification introduced the video element for the purpose of playing videos, partially replacing the object element. Clients can connect and stream videos conveniently using their computers, mobile phones and tablets.

4 Results and Discussion

4.1 Implementation of SAHVISS

The design was implemented using HTML5, PHP, Angular JS, Java Script, CSS with SQL server database. The program can run successfully on any system with the following properties: Windows XP Professional edition with at least 32 bit operating system, Windows Server 2008, Wamp Server 2.0, Dreamweaver and Code Editor Notepad ++ 3.0 and above or another equivalent e.g., PHP Storm

Fig. 2 shows the Home Page of the system while Fig. 3 shows the slide options to explore through the home page.

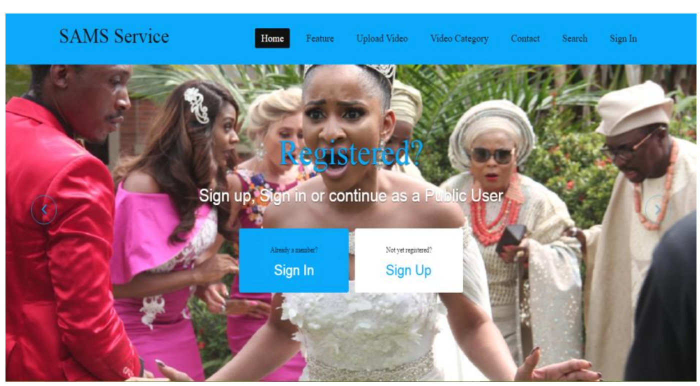

Fig. 2. SAHVISS home page

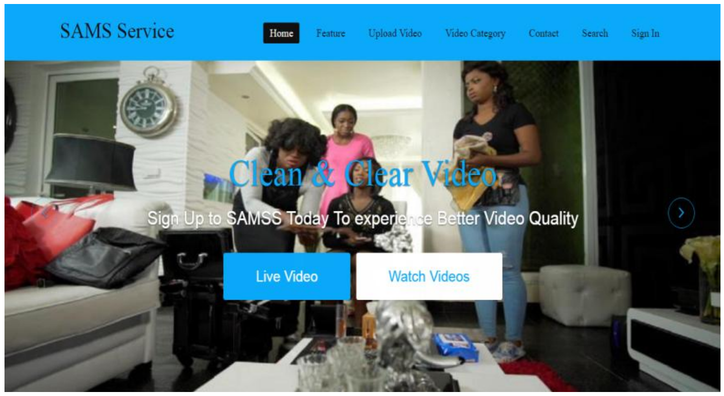

Fig. 3. Exploring the slide option of SAHVISS home page

Fig. 4 shows the video upload page for Registered users, the difference here is that registered users can upload up to 512mb video while non-registered users are limited to uploading up to 256mb video, also a registered member can set privacy for each video uploaded while a non-registered user cannot.

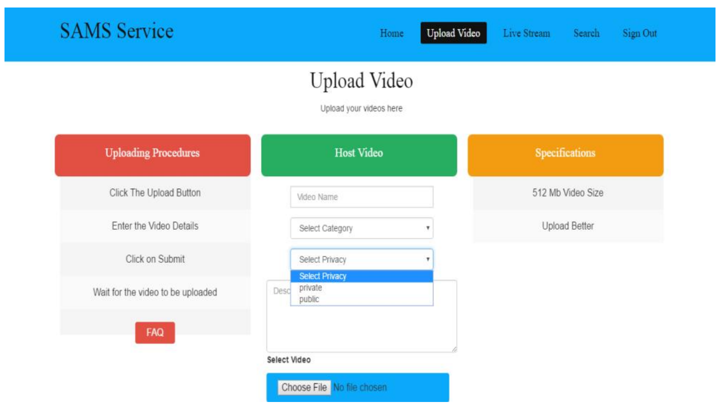

Fig. 4. SAHVISS video upload page for registered users

Fig. 5 shows Video Category page where users can limit their interest to their choice while Fig. 6 shows the view after clicking on all videos, SAHVISS brought out videos that are from all the category i.e. Educational, Discovery, Entertainment, Seminar etc.

Fig. 5. SAHVISS video category page

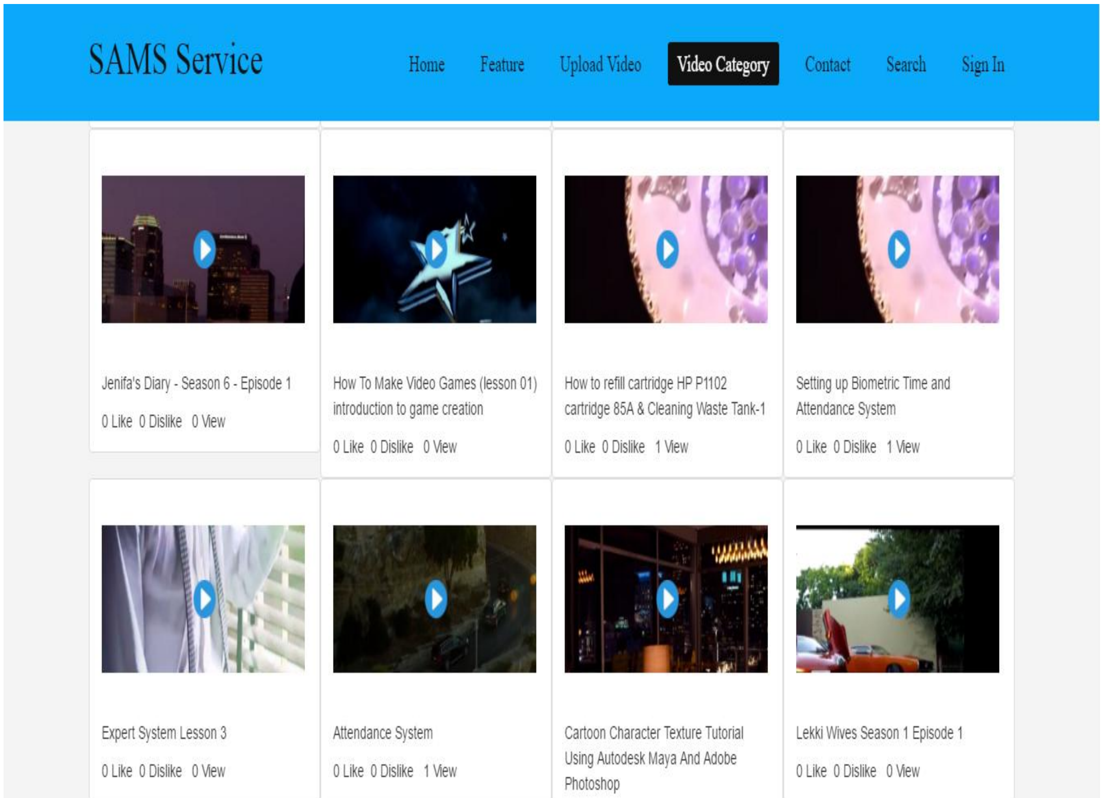

Fig. 6. SAHVISS all video page

Fig. 7 shows a video being viewed on SAHVISS, taking note of the buttons such as play, pause, stop, download, replay, end, like, dislike, and comment carrying out stated functionalities. Also below these buttons are the video name and the description of what the video is all about.

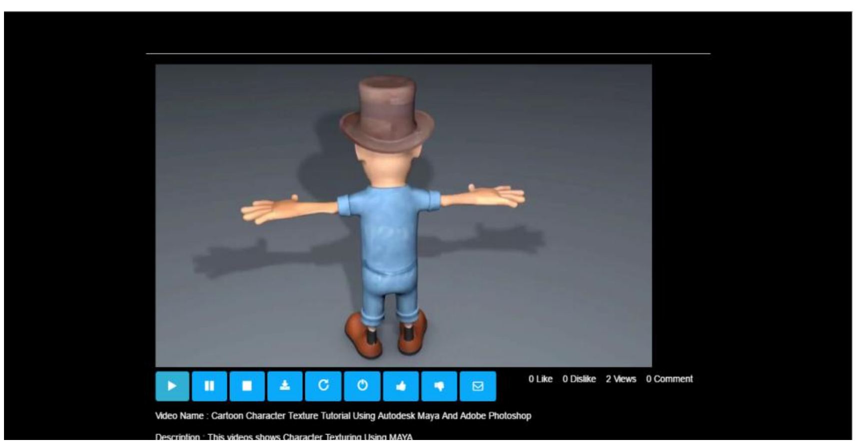

Fig. 7. Screen Snap of video being viewed on SAHVISS

Fig. 8 shows a series of comments already being made on that video by users.

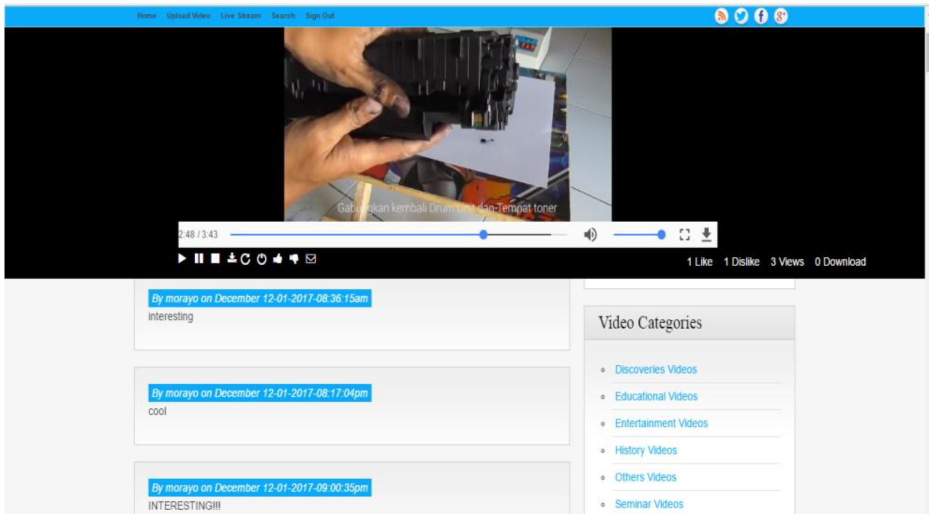

Fig. 8. Screen snap of comments made on a SAHVISS video

Fig. 9 shows side displays trending videos on SAHVISS, trending videos are videos with the highest number of views.

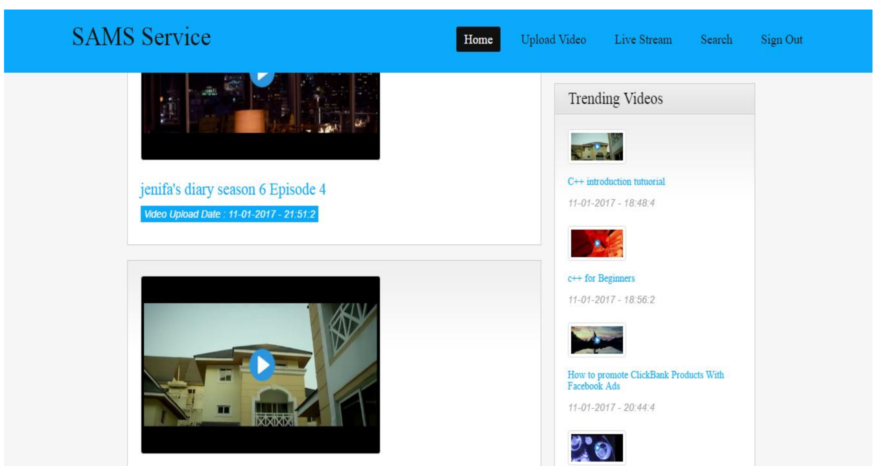

Fig. 9. Page showing trending videos on SAHVISS

5 Conclusion

In this paper, a secure and adaptive http video streaming system (SAHVISS) was designed. The system is an interactive web-based video streaming service that allows users of various walks of life to share and watch video in real time without necessarily downloading the video. With SAHVISS, video is watched almost immediately after being clicked.

The system allows users to set the privacy of video; the privacy can be set to either public or private to prevent unwanted access to posted/shared video. The system allows videos to adapt easily to user's network. The adaptive mechanism in this design performs two functions; it enables video to adapt to network changing conditions, and also to competently select the next video segment. The system allows users that want to upload the video to give a description of the video uploaded and also sort the videos in the appropriate categories.

In future, it might be necessary to develop better algorithms for a server-client streaming system that can support adaptive video streaming. Efficient layered video schemes such as H.264 SVC will allow a better adaptation to heterogeneous peers.

Competing Interests

Author has declared that no competing interests exist.

References

[1] Hiroki O, Hiroyuki H, Hiroshi N. Design, implementation and evaluation of congestion control mechanism for video streaming. International Journal of Computer Networks & Communications (IJCNC). 2011;3(3).

[2] Asghar M, Sadaf S. SVS - A secure scheme for video streaming using SRTP, AES and DH. 2010;177-188.

[3] Holankar D. Secure streaming media and digital rights management. International Technical Paper. 2011;1-4.

[4] Apostolopoulos JG, Tan W, Wee SJ. Video streaming: Concepts, algorithms, and systems. Mobile and Media Systems Laboratory, HP Laboratories Palo Alto; 2002.

[5] Mundhada RA, Khapre SS, Fulare P. An approach for sharing secured and efficient video streaming over the cloud. International Journal on Recent and Innovation Trends in Computing and Communication. 2015;3(5).

[6] Zhu X, Girod B. Video streaming over wireless networks. IEEE 15th European Signal Processing Conference; 2007.

[7] Saremi TJ. U.S. Patent No. 9,584,556. Washington, DC: U.S. Patent and Trademark Office; 2017.

[8] Bailey JM. Live video streaming from android-enabled devices to web browsers; 2011
Available:http://scholarcommons.usf.edu/etd/299

[9] Sabu MT. A review on P2P video streaming. Cornell University Library; Arxiv.org>cs>arXiv:1304. 2012;123.

[10] Pradan NL. Streaming video over the internet and wireless IP networks. The 2012 Annual Technical Conference & Exhibition. 2012;15:550-588.

[11] Patil J. Secure media streaming for android. A Project Presented to the Faculty of the Department of Computer Science San Jose State University; 2012.

[12] Junchen J, Vyas S, Hui Z. Improving fairness, efficiency and stability in HTTP-based adaptive video streaming with FESTIVE. SIGCOMM Conference; 2012.

[13] Kim Y, Shin J, Park J. Design and implementation of a network-adaptive mechanism for HTTP video streaming. ETRI Journal. 2013;35(1):27-34.

[14] Oda H, Hisamatsu H, Noborio H. Design, implementation and evaluation of congestion control mechanism for video streaming. International Journal of Computer Networks & Communications (IJCNC). 2011;3(3):139-204.

[15] Krishnamoorthi V. Efficient HTTP-based adaptive streaming of linear and interactive videos. Dissertation, Department of Computer and Information Science, Link¨oping University, Sweden; 2018.

[16] Manasa V, Vikram M. A secured adaptive mobile video streaming and efficient social video sharing in the clouds. International Journal of Computer Science and Information Technologies. 2014;5(4): 5153-5156.

[17] Brookins NS, Ludin SL, Olugbile AO, So R. U.S. patent no. 9,549,022. Washington, DC: U.S. Patent and Trademark Office; 2017.

[18] Shuai Y. Dynamic adaptive video streaming with minimal buffer sizes. Dissertation 2018.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
http://www.sdiarticle4.com/review-history/46761