INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT (IJSREM)
VOLUME: 05 ISSUE: 05 | MAY - 2021 ISSN: 2582-3930

VIDEO STREAMING THROUGH PEER-TO-PEER CONNECTION OVER THE CLOUDNETWORK

Ass Prof Rubana khan, MustafaYusuf Karanjawala, Deepali Bhoyar, Raghav Agasti,

Deepali Mahajan

ABSTRACT - In the video streaming platform, streaming is done either on the cloud platform or by the OBS online streamer such as the social YouTube streaming, twitch etc. In this paper we are going to discuss and give a torchlight to a new idea of streaming. We are going to discuss about the Peer-to-peer Streaming over the cloud network. Our streaming should not belong to any third-party streamer platform but only to the host and its peer. After terminating the stream by the host, streaming video should be deleted and terminated by the user who accessed it. It will increase the synchronization, security of their stream. We are going to read and understand the new concept of sharing the stream over the cloud network

Keywords *- Bitrate adaptation algorithm, peer to peer network streaming, QoE , RTMP server, cloud network*

Introduction

This paper is a collaborative, cloud-based videoconferencing service offering feature of Peer to Peer over the cloud Network as a single host for one peer to another peer network, that includes online meetings, group messaging services, and secure recording of sessions compared with platforms like Skype. It offers the ability to communicate in real time with geographically dispersed individuals via computer, tablet, or mobile device. However, unlike many other VoIP technologies, it possesses a number of additional advantages that enhance its potential research utility. A key advantage is its ability to securely record and store sessions without recourse to third-party software such as the YouTube, twitch etc. Why we use this platform? why can't we do the streaming by just sending and receiving the call while sharing the encrypted links? This paper will explain all the major and minor facts and how it can be achieved in a better way. This feature is particularly important in research where the protection of highly sensitive data is required. Other important security features include user-specific authentication, real-time encryption of meetings, and the ability to backup recordings to online remote server networks ("the cloud") or local drives, which can then be shared securely for the purpose of collaboration. The possibility that VoIP technologies can improve researchers' and participants' experiences of qualitative data collection is yet to be validated. The merits and shortcomings of VoIP technologies, as well as comparisons with in-person data collection, are typically based on researchers' subjective assessments of the quality of interview data. However, as Weller (2015) argues, "participants' experiences and assessments of the process have received far less attention" thus hindering efforts to improve interview quality and explore novel research applications. Further, most of the literature has concentrated on asynchronous online discussion forums rather than synchronous interaction.

We have parted this paper into four section where in first section we have the full discussion about Quality of experience (QoE)*which is the major characteristic of our approach. In this section we are going to discuss about QoE and its factor of streamingover the cloud network. Second section is about, why we are going for these approaches? and what is the objective behind this approach? as we already have the zoom and google meet application services which performs the same features so how is this different from it? and all the factor

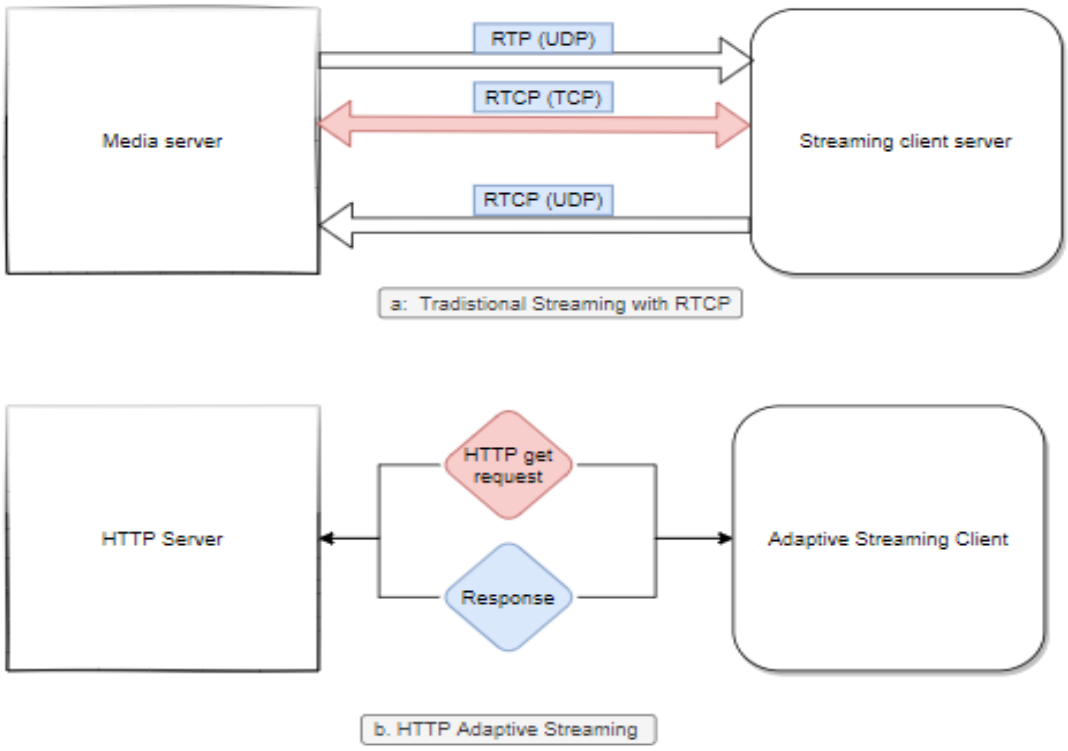


Fig: 1 Adaptation of streaming over internet

related to it. In third section we are going to discuss about, what major actions can be taken under consideration to overcome the drawbacks of this approach.

Fourth section is of Software requirement, i.e., what software is required and how we can use this software to achieve our approach of peer-to-peer streaming, we are going to be discuss about the changes that can take in this software's.

It offers the ability to communicate in real time with geographically dispersed individuals via computer, tablet, or mobile device. However, unlike many other VoIP technologies, It possesses a number of additional advantages that enhance its potential research utility.

CONCEPTS UNDER CONSIDERATION.

A. Bitrate adaptation Scheme for streaming media over HTTP.[17]

Video conferencing has evolved to major factor on internet world. Trend pose challenges in the delivering with the best QoE over the todays Internet world, which was originally designed for best-effort, non-real-time data transmission.

Around 2005, an elegant yet simple video delivery paradigm was introduced by Move Networks, which quickly became popular due to its better features and cheaper deployment costs over progressive download and other proprietary streaming methods. This new paradigm, which we refer to as HTTP adaptive streaming (HAS), treated the media content like regular Web content and delivered it in small pieces over HTTP protocol.[1][17]

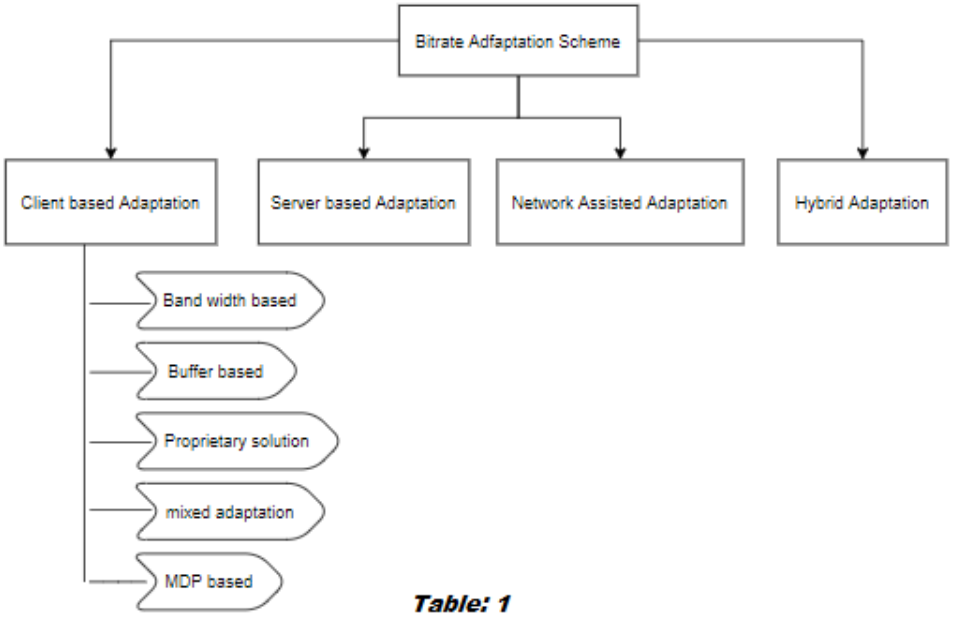


Table: 1

In traditional non-HAS IP-based streaming show in the fig:1 a, the client receives media that is typically pushed by a media server using either connection-oriented protocols such as the Real-time Messaging Protocol (RTMP/TCP) [2] or connectionlessprotocols such as the Real-time Transport Protocol (RTP/UDP) [3]. A common protocol to control the media servers in traditional streaming systems (asa shown in Fig. 1a) is the Real-time Streaming Protocol (RTSP) [4]. RTSP is responsible for setting up a streaming session and keeping the state information during this session, but is not responsible for actual media delivery, which is thetask for a protocol such as RTP. Based on the RTP Control Protocol (RTCP) reports sent by the client, the media server may perform rate adaptation and data delivery scheduling. These characteristics result in complex and expensive servers. Additional protocols or configurations areneeded during the session establishment in case network address translation (NAT) devices and firewalls block the control or media traffic [5]. Despite implementing the same baseline protocol(s), media servers from different vendors may behave differently due to optional features or differences in implementation. Failovers due to a server fault often cause presentation glitches and are rarely seamless unless certain redundancy schemes are in place. These scalability and vendor dependency issues as well ashigh maintenance costs have resulted in deployment challenges for protocols like RTSP.

In current discussion survey, the author classified the schemes into three main categories: In Network approaches, server side and client-side approaches. They provided a general review of the video traffic and a set of factor and characterization studies for well-known commercial streaming provide in Ott platform such as the YouTube, twitch etc. This research on the Http over the stream involves to major aspects.1) that is more and more schemes should be examine and should be done under details compressiontable [1].2) classification of the stream must be provided that has the unique feature for adapting the logics.[1]

In the HAS solution sown in fig:1 a integration most of the integrate the bitrate adaptation logic inside the its client, since it allows the client to select a bitrate level independently which lead to avoid therequirement of having intelligent devices in our technologies. This are the key solution why the http is mostly use in the OTT platform. For instant client may use streaming server or any server or the network in the bitrate adaptation to improve the level of quality of Experience (QoE). By using IP

multicasting t simplifies the video distribution while connected devices. Classification of the bitrate can be shown by below diagram as well. Get the bitrate adaptation schemes to give the access in our main objective of the paper.

While Moving from a server-push to a Peer-to-peer-pull model has clear benefits, Still faces the challenges. Related issues that increase the number of users, and the growing demand of the high-quality content and stabilize content. Author discusses the main three problems that can affects these issues are: 1) Multiple-client competition issue, 2) consist streamingQuality, 3) Quality of Experience optimization, 4) Inter-destination multimedia Synchronization.

1) *Multimedia client compatibility Issue* [6]: to enhance the video Qo there are the three main objectives taken into the consideration. That are :1) Stability: server clients should avoid frequent bitrate switching, which leads to quality oscillations and video stalls, which in turn can negatively affect QoE. • Fairness: Multiple HAS clients competing for available bandwidth should equally share network resources based on viewer-, content-, and device characteristics. The fairness desired here does not often result in bandwidth fairness. • High Utilization: While the clients attempt to be stable and fair, network resources should be used as efficiently as possible.

A streaming consists of the two factor the buffer state and the steady state the buffer-filling state aims to fill the playback buffer and reach a certain threshold where the playback can be initiated or resumed [7]. In this state, the client requests the next segment as soon as the previous chunk is fully downloaded. After the playback buffer level reaches threshold, the client enters in the steady state.

2) *Consistent-Quality Streaming*: according to the research and analysis [8], [9],confirmthat the corelation between the video bitrate and the video frame are nonlinear in nature. Additionally, different video content types have unique characteristics that are the motion high level and low level which main leads to consistent action.

3) Changing condition of the best sentnetwork interduce numerous problems in delivering the multimedia viewer. In traditional non adaptive streaming, the client steams a video that is typically available in bitrate at the server side. If the network condition is worser in the case of downloading rate may fall below the play back rate. Which may lead to buffer. This approach may lead to the bitrate adaptation algorithm the part which may responsible for the downloading.[1]

4) *Inter destination multimedia Synchronization:[11],[12]:* Apart from online gaming, photo sharing, and instant messaging, online communities are drifting towards watching online videos together in a synchronized manner. Having multiple streaming clients distributed in different geographical locations poses challenges in delivering video content simultaneously, while keeping the playback state of each client the same (playing, paused). Moreover, it becomes more challenging for HAS streams to synchronize, since each client adaptively streams depending on their current network conditions. This problem is called Inter-Destination Multimedia Synchronization (IDMS). Typically, IDMS solutions involve a master node (either a dedicated master or a peer among the streaming clients in a session) to which clients synchronize their playout to. One of the earliest papers in this field was published by Montaged et al. [10-46], in which the authors discuss use cases where IDMS and its schemes is essential. Rainer et al. [11], [12] proposed an IDMS architecture for DASH by using a distributed control scheme (DCS) where peers can communicate and negotiate a reference playback timestamp in each session. The MPD file was altered to include IDMS session objects that enabled session management. In another

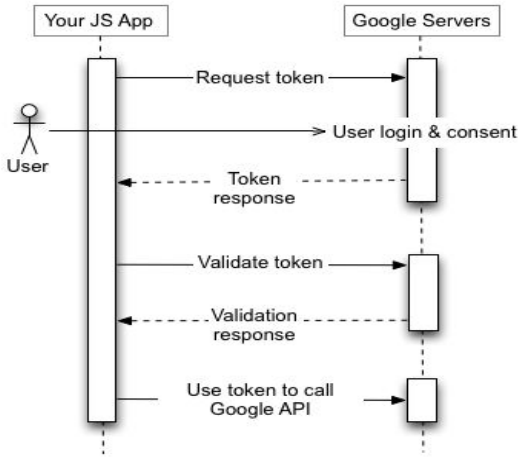
work [49], Rainer et al. provided a crowdsourced subjective evaluation to find an asynchronism threshold at which QoE was not significantly affected. They found that an asynchronism level of 400 ms was acceptable compared to the synchronous reference case. Synchronization in IDMS systems is crucial to the QoE. Dedicated QoE models have to be developed that take the visual quality, user engagement and the synchronization accuracy into account.

Now this all section is described because to explain the bitrate algorithms we are already going to use in the streaming platform. Further, we will discuss about the Authentication requirement in our approach. Authentication requirement in ourapproach is simpler, by **using** like google OAuth2 platform.

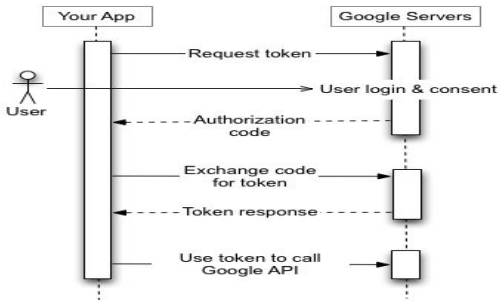
GOOGLE OAUTH 2.0 *[13],[14],[19],* is the authorize channel present by the google for auth theapplication or any software which may require only public data to access. We are handling and taking only that considerable data within only public by the google itself such as the Gmail id, Gmail name, Gmail phot etc actually we can only allow the public ally expose data to access in our approach. As there is compulsory for us to give the auth in our software as because if it will be complete public then hacker may access the data over while. Now will explain the auth platform in the simple three steps: that are

1) *Cross-client Identity*: When developers build software, it routinely includes modules that run on a web server, othermodules that run in the browser, and others that run as native mobile apps. Both developers and the people who use their software typically think of all these modules as part of a single app.

2) *Cross-client Authorization Goals*: When an app uses OAuth 2.0 for authorization, the app acts on a user's behalf to request an OAuth 2.0 access token for access to a resource, which the app identifies by one or more scope strings. Normally, the user is asked to approve the access 3) Cross-client access tokens Software can obtain OAuth 2.0 Access tokens in a variety of ways, depending on the platform where the code is running. For details, see Using OAuth 2.0 to Access Google APIs. Normally, user approval is required when granting an access token. OAUTH 2.0 allows users to share specific data with an application while keeping their usernames, passwords, and other information private. For example, an application can use OAuth 2.0 to obtain permission from users to store files in their Google Drives. This OAuth 2.0 flow is called the implicit grant flow. It is designed for applications that access APIs only while the user is present at the application. These applications are not able to store confidential information. In this flow, your app opens a Google URL that uses query parameters to identify your app and the



type of API access that the app requires.



You can open the URL in the current browser window or a popup.

Fig: 2 Case diagrams for google auth

The user can authenticate with Google and grant the requested permissions. Google then redirects the user back to your app. The redirect includes an access token, which your app verifies and then uses to make API requests. And the third thing is about the Realtime server exercise.

REAL TIME VIDEO QOE ANALYSIS OF RTMP SERVER

Real-Time Messaging Protocol (RTMP) [16] is a protocol for streaming audio, video, and data over the Internet. RTMP is a TCP-based protocol designed to maintain low-latency connections for audio and video streaming. To increase the amount of data that can be smoothly transmitted, streams are split into smaller fragments called packets.

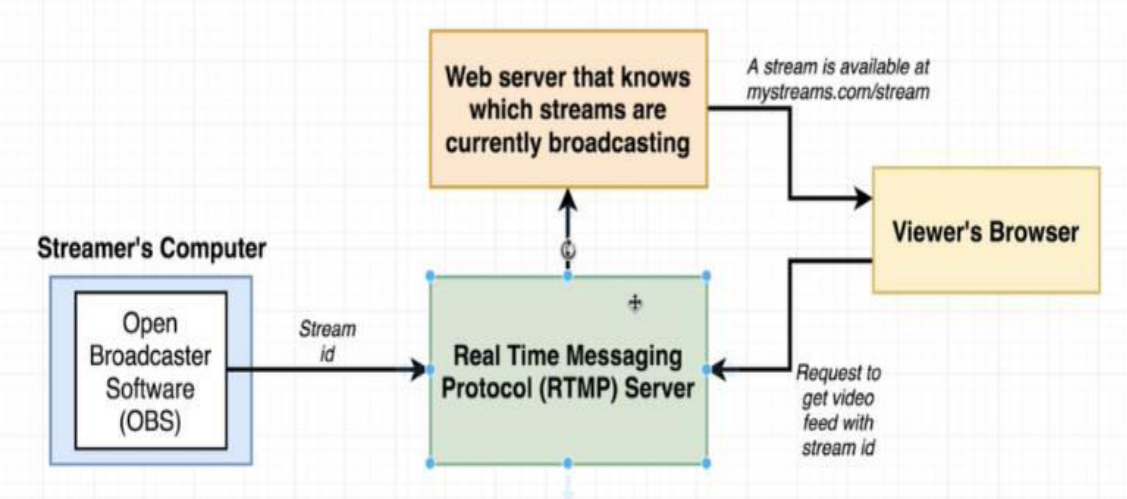


Fig: 3 Management flow of RTMP server in any software

RTMP also defines several virtual channels that work independently of each other for packets to be delivered on. This means that video and audio are delivered on separate channels simultaneously.

At a high level, Real-Time Messaging Protocol works as a three-step process.[16],[17]

Step 1: The handshake

Once RTMP establishes a TCP connection it performs a handshake by exchanging three packets between the client and the server. These packets are called chunks. The process looks like this: 1) The client sends over a chunk to tell the server which protocol version it is using. 2)Without waiting for a response, the client sends another chunk with a timestamp.3) The server responds with an echo of the chunks it receives, this time including a timestamp of the time it received them.4) The client sends one final packet that is a copy of the timestamp packet and the server returns it.5) Once the final packet exchange is complete, the handshake is considered complete.

Step 2: The connection

The client and server can now begin negotiating a connection through Action Message Format (AMF) messages. And the server will receive the request and respond with the appropriate message sequence.

Step 3: The stream

The client can now start a video or audio stream by sending three messages to the server: create Stream, ping, and play.

METHOD OF THE PEER-TO-PEER STREAMING NETWORK

For this, we are going to take simple example and try to give the brief in it. Now there is the person's name X and it's a Saturday night, x wants to watch movie with x friends, but unfortunately his friends are out of the station or maybe there is the pandemic going on or whatever the reason. But x wants to watch the movie with his friends. Now, our approach comes into the plan, x will have the movie in HDD storage in the local environment, x will open our software (peer to peer streaming software), and upload the movie in ourprovided cloud. After that x will create new stream and then drag the file from the cloud to start the movie, before starting the movie our software will generate the encrypted link, that link will be share by x to his friends who want to watch the movie. After sharing the link x will start watching the movie in his own personal server. While watching the movie, other friends will also start watching the same movie in the same time interval, using same platform online but without any interruption in playing the movie, which is synchronized by the x as a hosted to the single server. that is pretty much it. It is simple, so now we are going to talk about some important terms and concept that are pointed by us in previous example:

Firstly, when x will upload the movie into the cloud provider by our software that means x read that uploaded file with the help of media player, which is already being installed by the cloud provider. Basically, x will watch the media file from its cloud. Now, while accessing the media file, x's server is reading that media file via media player (any media player software which is already been installed) i.e., reading the frames of media file that is running on the media player software, which is the main objective of watch any media file, any multimedia content. X have his own personal cloud and from there he is watching, same as we upload media file into google-drive and then watch from drive without any conceptional thoughts.so what is the difference? Difference, is that while playing the movie, instead of running it into the local environment, the media file is running directly as a stream. Will discuss it later about the listening and encrypted link concept in the section. Now, where the server has been created? When x has uploaded the media file into the cloud of our platform, then automatically the server has been created, and that server is the main server that has been created by the x. That server is being carried forward and watched by x's friends.

Now, Listening and listener concept, the person who is the hosted will create the server to host any media file, upload it in the cloud and then start that media file, this is called as the user devise i.e. listening the call from the multimedia access (multiplayer software). After that, user host will share the encrypted link to his friends, and after accessing the host file they all are the listener, because they are listening the call of the host via encrypted link, which is actual running the file from the cloud. Peer to peer concept is like the listening to the listener concept. We are not using our software approach as a third-party approach like google drive or zoom service provider does, i.e., by generating links and then data will be passed over the company's network to the client network. Here we are connecting them in-between their local environment same as the concept of block chain network. Like blockchain architect works in the same way our streaming would work, no centralize authority should be there, nothing should be taken in the consideration only an object will connect to another object by knowing there IP address and IPv4 address, which should be mandatory to recall for data security. Now, will discuss about the listeners, x's friend to whom x will be sharing the link. That link contains the reading media files, which are currently reading the uploaded media file that are being currently seen by x . So, apparently, we can say that listener is not actually watching the media file but their link is reading the synchronization of media with x server. They are not actually watching the media from the hosted server but listening the server – user host network, which is publicly over the browser, hosting will be allowed by creating the proxy servers to their

listeners to access the data from its hosted device. Hope you are getting the main concept clearly.

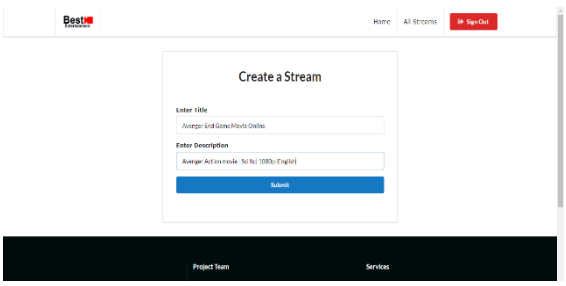
If host started the streaming and then share the encrypted link, the listener who are supposed to join it, they will join in between the steam and they will not have any right to forward or backward or paused the media file. Listener only can watch just like the broadcasting done at TV. In T.V broadcasting is done though the satellite as the central authority, but that is not the case in our streaming, in our idea, the connection is in between the computer devices. So, host can only forward or backward the file which is belong to host.

After completing streaming over then the host will terminate the stream, the two-factor come into consideration while terminating the server: 1) the media file should be hold in the cache of server 2) all the belonging to host can be terminating after host cancel the stream.

When the host terminate the stream after completing it, in first case we are going to put the record of their stream in the cloud storage. Host will be deleted from the software, but only key id will be taken into database, for recognizing the user, after when it login again. In this scenario main drawback is for the listeners. Suppose the listener come to the link and open it(Link will be ON as the media file is still in the cloud storage) , but listener will get it as media is currently running in the server(Actually not accessible). The user listener will drain a large amount ofdata unnecessarily without actual stream showing. Now why it is happening? because, when the streaming is live on with the host network, then the listener role is to watch the call from host server and cloud media, not directly the media, but now after the user host terminate the stream from the software but not delete the media file, still listener is in the encrypted link(as the link is not terminated) and link is ON, now they will start directly reading the file from the cloud as a result they will read the media file from their local environment and start draining the large amount of internet data. As we earlier explain that listener will never eligible for any too and fro in media file so this is use less for draining the data internet from their devices.

Now for these we have the second approach which neglect the first drawback. When user host terminate the stream, it will delete all over from browser including from the cloud cluster. And link will get expire immediately. Listener will be thrown out from the browser link. User host ID will be vanished from the software. Now, when the same user Login again, then he will be considered as a new user host and will start from the first step, will upload the media file, then start the stream and share the encrypted link. This approach will neglect the major drawback from the software which has been discussed by us. Now, this also has minor drawback, that user host will not able to have the preferable options such as downloading the stream form its cloud once it will upload, and would not know how many users have been attached after terminating the field. But these minor drawbacks can be neglected by enhancing the software.

SCREENSHOTS OF OUR EXPERIMENTAL WEB APPLICATION SOFTWARE.



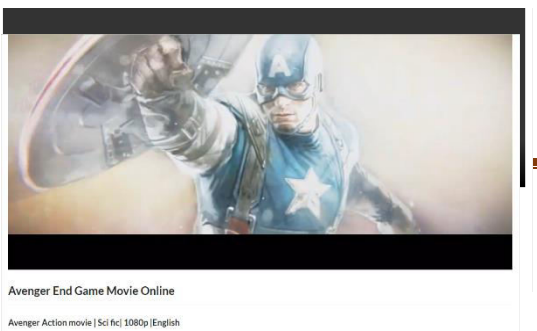
Stream creation:*Screenshot 1*

Edit and delete the stream:Screenshot 2

Streaming is running:

Screenshot 3

FUTURE SCOPE AND APPLICATION AREA.



Streaming over peer-to-peer network as a single hosting server.

There are challenges that are unique to running a small business — executing multiple processes with just a few people and limited budget, and prioritizing the time and resources to make it all happen efficiently — that can be solved with this platform. Why? Because it provides easy-to-use video communications that empower people to accomplish more. Communications should help every growing business achieve its goals, but also help meet its constantly evolving needs. Zoom's simplicity and ease of use, not to mention its tiered pricing and usage plans, simplify how small-business teams manage their time, enhance productivity, and scale the company.

We're all about flexibility and simplicity. This web Application works seamlessly across all your operating systems — PC, Mac, Linux, iOS, and Android — so your employees aren't locked into specific devices. You also can affordably video-enable any conference room or meeting space, which also are hardware agnostic and simple to set up (even first-graders can do it!). Use your resources wisely with a single solution for meetings, webinars, phone, and chat. Your employees only have to use one tool, and they'll love you for that! The best part: This continues to innovate its platform to meet expanding business needs, so you'll never have to worry about finding another communications platform.

This project proposes the use of Social Live Streaming Tools in various devices in order to facilitate the development and is built to let dozens of people join the Gathering with each other from anywhere with internet access. Unlike other online meeting medium the project serves availability and accessibility to large group of people for longer duration. Full access will be given to host who can share files (audio, video, data record, images) or organise live conferences. Host will have access to bifurcate people according to their priority and divide them into certain rooms with different features People with higher priority can interact with audio, video and chat facility while other will interact via live video streaming.

CONCLUSION

We have successfully discussed the idea and concept behind the Peer-to-peer streaming over the cloud network. We have discussed in this paper about the streaming bitrate algorithm and how it will affect in the video streaming over the peer-to-peer network. We can successfully authenticate with the application via google OAuth 2.0 and have full security over our data in the application. Listener will listen the call of video streaming successfully, which has been shared by the owner of the stream via our idea of streaming. Listener doesn't require the application for listening the stream from their parent link. Can successfully watch stream without actually application downloaded. We have discussed in this paper about the approach taken by the host network to overcome the large amount of data drain over the internet which is not required. Listener doesn't require the application for listening the stream from their parent link. Can successfully watch stream without actually application download. Video conferencing can successfully do on the single hosting platform by the owner of the file and can be listened in social network area.

REFERENCES

[1] "Cisco visual networking index: Forecast and methodology, 2016– 2021," San Jose, CA, USA, Cisco Syst., Inc., White Paper, 2017.

[2] Real-Time Messaging Protocol (RTMP), Adobe, San Jose, CA, USA, 2014. Accessed: Nov. 21, 2017. [Online]. Available: http://www.adobe.com/devnet/rtmp.html

[3] V. Jacobson, R. Frederick, S. Casner, and H. Schulzrinne. (2014). RealTime Transport Protocol (RTP). Accessed: Nov. 21, 2017. [Online]. Available: https://www.ietf.org/rfc/rfc3550.txt

[4] H. Schulzrinne. (2016). Real Time Streaming Protocol Version 2.0. Accessed: Nov. 21, 2017.

[Online]. Available: https://tools.ietf.org/html/rfc7826

[5] J. Goldberg, M. Westerlund, and T. Zeng. (2014). A Network Address Translator (NAT) Traversal Mechanism for Media Controlled by the Real-Time Streaming Protocol (RTSP). Accessed: Nov. 21, 2017. [Online]. Available: https://tools.ietf.org/html/rfc7825

[6] M. Seufert et al., "A survey on quality of experience of HTTP adaptive streaming," IEEE Commun. Surveys Tuts., vol. 17, no. 1, pp. 469–492, 1st Quart., 2015.

[7] S. Akhshabi, S. Narayanaswamy, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptive video players over HTTP," Signal Process. Image Commun., vol. 27, no. 4, pp. 271–287, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0923596511001159

[8] G. Cermak, M. Pinson, and S. Wolf, "The relationship among video quality, screen resolution, and bit rate," IEEE Trans. Broadcast., vol. 57, no. 2, pp. 258–262, Jun. 2011.

[9] C. Kreuzberg, D. Poach, and H. Hellwagner, "A scalable video coding dataset and toolchain for dynamic adaptive streaming over HTTP," in Proc. 6th ACM Multimedia Syst. Conf. (MMSys), 2015, pp. 213–2

[10] M.Montagud, F. Boronat, H. Stokking, and R. van Brandenburg, "Inter-destination multimedia synchronization: Schemes, use cases and standardization," Multimedia Syst., vol. 18, no. 6, pp. 459–482, 2012.

[11] B. Rainer and C. Timmerer, "Self-organized inter-destination multimedia synchronization for adaptive media streaming," in Proc. 22nd ACM Int. Conf. Multimedia (MM), 2014, pp. 327–336. [Online]. Available: http://doi.acm.org/10.1145/2647868.2654938

[12] B. Rainer, S. Petscharnig, and C. Timmerer, "Merge and forward: Selforganized inter-destination multimedia synchronization," in Proc. 6th ACM Multimedia Syst. Conf. (MMSys), 2015, pp. 77–80. [Online]. Available: http://doi.acm.org/10.1145/2713168.2713185

[13] M. Darwish and A. Ouda, "Evaluation of an OAuth 2.0 protocol implementation for web server applications," *2015 International Conference and Workshop on Computing and Communication (IEMCON)*, 2015, pp. 1-4, doi: 10.1109/IEMCON.2015.7344461.

[14] Ferry, E., O Raw, J. and Curran, K. (2015), "Security evaluation of the OAuth 2.0 framework", *Information and Computer Security*, Vol. 23 No. 1, pp. 73-101. https://doi.org/10.1108/ICS-12-2013-0089

[15] H. French, J. Lin, T. Phan and A. C. Dalal, "Real time video QoE analysis of RTMP streams," *30th IEEE International Performance Computing and Communications Conference*, 2011, pp. 1-2, doi: 10.1109/PCCC.2011.6108105.

[16] P. Zhao, J. Li, J. Xi and X. Gou, "A Mobile Real-Time Video System Using RTMP," *2012 Fourth International Conference on Computational Intelligence and Communication Networks*, 2012, pp. 61-64, doi: 10.1109/CICN.2012.18.

[17] A. Bentaleb, B. Taani, A. C. Begen, C. Timmerer and R. Zimmermann, "A Survey on Bitrate Adaptation Schemes for Streaming Media Over HTTP," in *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 562-585, Firstquarter 2019, doi: 10.1109/COMST.2018.2862938.

[18] B. Rainer, S. Petscharnig, C. Timmerer, and H. Hellwagner, "Is one second enough? Evaluating QoE for inter-destination multimedia synchronization using human computation and crowdsourcing," in Proc. 7th Int. Workshop Qual. Multimedia Exp. (QoMEX), 2015, pp. 1–6.

[19] https://developers.google.com/identity/protocols/oauth2