INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

IJARIIT

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 3)
Available online at: www.ijariit.com

Survey on low latency live video streaming

Priyanka Deshmukh
deshmukh.priyanka23@gmail.com
K J College of Engineering and Management Research, Pune, Maharastra

Nagaraju Bogiri
mail2nagaraju@gmail.com
K J College of Engineering and Management Research, Pune, Maharastra

ABSTRACT

Video gushing over Internet has got bunches of consideration as of late. These days over half of the worldwide information activity is devoured by video bundles and it will be over 80% by 2020. Live gushing is testing, since it needs ongoing techniques with low dormancy. Advanced mobile phones and tablets are the new age of PCs with the capacity to do some of our days by day schedules. Live spilling between two cell phones has numerous applications, for example, reconnaissance, video talk, and so on. In this paper we proposed a technique to stream live video from a cell phone to another, utilizing a web-socket. For assessment, we have executed an open source library on Android, being used for any individual who needs to utilize live video gushing as a piece of their application, and showed that our strategy can play remote video with lower than 2 seconds delay in various situations. Moreover, our technique increment and decrement latency according to network condition so as to give a superior nature of experience to the viewer.

Keywords: *Low latency, Quality of Service, Long Term Revolution, WebRTC*

1. INTRODUCTION

Ongoing Communication (RTC) over voice and video has a few advantages, yet because of a few issues, for example, costly video and sound authorizing, RTC represents a few difficulties that have pulled in the exploration group [5]. The World Wide Web Consortium (W3C) and Internet Engineering Task Force (IETF) built up another standard known as WebRTC; they have remarked that the WebRTC is intended to allow the co-event of sound and video sessions without the need to modules or different expenses. WebRTC is a shared open source structure that is considered as an accumulation of guidelines, conventions, and JavaScript [6]. Additionally, it is upheld by Opera, Mozilla Firefox, and Google Chrome. Section 2 introduces the related work on adaptive video streaming. Section 3 consists of the system model, problem formulation, and performance metrics. Results and discussion are presented in Section 4, and Section 5 consists of conclusion and future works.

2. RELATED WORK

In Arun Raj*, Dhananjay Kumar, H. Iswarya, S. Aparna and A. Srinivasan [1] paper a new system to support streaming of live and stored video through a wireless network is proposed which is based on adaptive playback buffer management on the top of HTTP at the client. The cushion completion is dealt with as an immediate state variable that mirrors the change of the system data transmission. The cradle totality estimation predicts the support status at a point later on in view of perceptions of the cushion over a stipulated timeframe.

Behin Molaei Tabari, Jafar Habibi, Abolhassan Shamsaie, Alireza Ma- Zloumiand Pedram Beheshti [2] proposed a method to stream live video from a mobile device to another one, using a web-socket. For assessment, they have actualized an open source library on Android, being used for any individual who needs to utilize live video spilling as a piece of their application, and showed that this strategy can play remote video with lower than 2 seconds delay in various situations. Besides this strategy increment and reduction inertness as per arrange condition keeping in mind the end goal to give a superior nature of experience to the watcher.

In Xu Na, Sun Shuang [3] paper, to improve the playback quality of P2P media streaming system terminal nodes and enhances the overall performance, a data scheduling algorithm(LDSA) is proposed, it is able to dynamically adjust the pending request according to the node ability. The algorithm in satisfies the media streaming living in the time response foundation had considered how to minimize the waiting time for the requests in the node and the rapid distribution in a network of scarce data blocks.

Naktal Moaid Edan, Ali Al-Sherbaz, Scott Turner [4] depicts the Web Real-Time Communication (WebRTC) innovation and the usage of its customers and server. The primary point is to plan and execute WebRTC video conferencing between programs in genuine usage utilizing Chrome and (Wired Wi-Fi) of LAN WAN systems. Additionally, an assessment of CPU execution, data transfer capacity utilization and Quality of Experience (QoE) was accomplished. Besides, a flagging channel between programs utilizing the Web Socket convention through Node.js stage has been made and executed. This paper gives web engineer a chance to appreciate the WebRTC innovation, and in addition to seeing how to plan WebRTC video conferencing.

3. IMPLEMENTATION

We built up an open source live video spilling library for android gadgets. Right now it utilizes Media Codec which is accessible in API levels 16 and higher, so it can keep running on gadgets running Jellybean or fresher ones. We utilize the H.264 Video and AAC Audio codec in light of the fact that they have great pressure proportion and are bolstered by most cell phones these days. Our strategy and codes don't rely upon these codecs specifically, so anybody can transform it in the Audio/Video Encoder and Audio/Video Decoder. Android Camera API and Audio Record were utilized to get the crude information from the gadget equipment. As we tried our application, some more established gadgets don't bolster some shading groups for the camera and encoder, this presents a few bugs in these gadgets; however more current models with KitKat and later were fine in our tests.

In our approach, we are utilizing a web-socket relay server to set up an association amongst source and goal gadgets. Realizing that most versatile systems put customers behind a NAT and they can't convey to each other specifically, the relay server must have a legitimate IP address and be noticeable on the Internet. One gadget goes about as the source and the other one goes about as a watcher. In Fig 1, you can see the entire framework in a look.

3.1 Relay Server

It has a web-socket server execution that transfers messages between two gadgets. Our relay server has a few likenesses with standard TURN server, it is utilized to make the in-coordinate association between two gadgets behind NAT. However our relay server utilizes web-socket for correspondence, and it distinguishes customers with their UUIDs.

3.2 Frame Handler

Frame handler has the most important part in our method. It has a less complex plan in source gadget when a video outline ends up noticeably prepared, it utilizes the present time in Unix age arrange as the time-stamp of that frame. On sound frames, it should include a suitable time-stamp to keep sound and video content in a sync. To mix the sound and video outlines in a similar stream, Frame Handler utilizes even timestamps for the video outlines and odd time-stamps for the sound casings.

3.3 Web Socket Client

Web Socket clients in source and watcher gadgets, handle correspondence with the relay server. The two customers build up an association with the server before video stream start. Watcher charges to Play or Stop experiences the relay server to the next gathering. Web attachment handlers ought to make a virtual Input/Output stream that goes about as the gushing stage amongst source and watcher Frame Handlers for video conveyance.

Our library has defined two states for the web socket client in the source device.

- **Waiting for Connection**, in this state web socket customer is associated with the relay server and it's sitting tight for *start stream* command from a watcher gadget.
- **Streaming**, in this state web socket customer, is currently spilling video substance to a watcher gadget. *Start stream* charges will be rejected with *source-busy* reaction. By getting *stop-stream* command, this customer changes back to Waiting for Connection state.

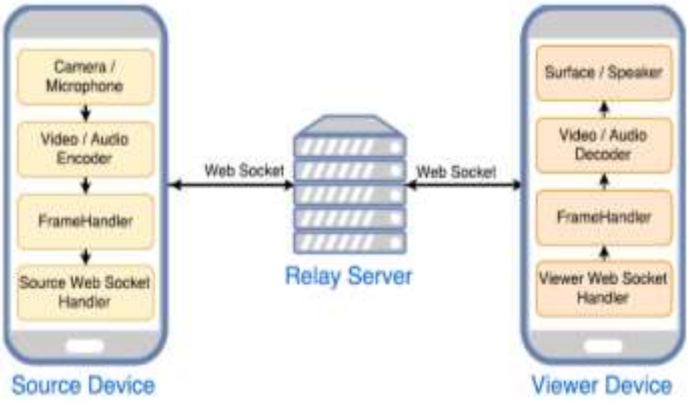

Fig. 1: Live Stream Video Architecture

4. RESULT

We have contrasted our gushing technique and some comparative strategies and this examination are accessible in Table 1. Versatile support is the capacity to change cushion measure in various system conditions. RTP and WebRTC do not have supports. On the off chance that poor system conditions delay the playback for a couple of moments, and it continues from that point forward, video inertness increments at that point. On the off chance that a strategy can play recordings speedier (or cut the video) to lessen the inactivity it has Frame time match up. RTP and WebRTC continue video from the most recent edges they get, so in their strategy video inertness is as low as could be expected under the circumstances, having some video cuts in these circumstances.

Table 1: Low latency live video streaming methods comparison

Method	Latency	Frame Time Sync	Adaptive Streaming	Protocol
RTP	<0.5 sec	Yes	No	UDP
WebRTC	<1 sec	Yes	Yes	HTTP and UDP
Wei and Swaminathan	<5 sec	No	Yes	HTTP 2.0
Cherif et. al	2 sec	No	Yes	HTTP 2.0 and Web-socket
Petrangeli et. al	2 sec	No	Yes	HTTP 2.0
Proposed Method	1.8 sec	Yes	Yes	Web-socket

5. CONCLUSION

This paper is proposed with the gushing live information among various clients utilizing WebRTC server. Also, helpful in video visits, sound talks among different clients. It can store the information in the database. Additionally, we proposed a low inertness spilling strategy to stream live video between two cell phones on the Internet. We additionally built up an open source library to be utilized as a part of android applications. Our technique needs a thin transfer server to permit aberrant associations between two gadgets behind any sort of NAT.

6. ACKNOWLEDGMENT

I avail this chance to express my profound feeling of appreciation and entire hearted on account of my guide Prof. Nagarju Bogiri Sir for giving his important direction, motivation, and consolation to set out this paper. Without their Coordination, direction, and surveying, this assignment couldn't be finished alone. Prof. Nagarju Bogiri sir gave me all the flexibility I required for this project.

7. REFERENCES

[1] Arun Raj*, Dhananjay Kumar, H. Iswarya, S. Aparna and A. Srinivasan Adaptive video streaming over HTTP through 4G wireless networks based on buffer analysis (2017) 2017:41DOI 10.1186/s13640-017-0191-4

[2] Behin Molaei Tabari, Jafar Habibi, Abolhassan Sham-Saie, Alireza Mazloumiand Pedram Beheshti: Low Latency Live Video Streaming on Android Devices using Web-Socket IEEE – 40222

[3] Xu Na, Sun Shuang Scheduling Algorithm used in Live Media Streaming Based on Data-driven Overlay Network Physics Procedia 33 (2012) 424-428

[4] Naktal Moaid Edan, Ali Al-Sherbaz , Scott Turner, Design and evaluation of browser-to-browser video conferencing in WebRTC, DOI:10.1109/GIIS.2017.8169813 Publisher: IEEE ISSN:2150-329X (2017)

[5] V. Swaminathan, Are we in the middle of a video streaming revolution? ACM Transactions on Multimedia Computing, Communications, and Applications, vol. 9, no. 1s, pp. 16, Oct 2013.

[6] S. A. Alvi, B. Afzal, G. A. Shah, L. Atzori, and W. Mahmood, Internet of multimedia things: Vision and challenges, Ad Hoc Networks, vol. 33, pp. 87111, 2015.

[7] Cisco, The zettabyte era: Trends and analysis, http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni hyperconnectivitywp.html, updated: 02/06/2016. 1683–1694 (2011)

[8] A. Bergkvist, D. C. Burnett, C. Jennings, and A. Narayanan, "Webrtc 1.0: Real-time communication between browsers," *Working Draft, W3C*, vol. 91, 2012.

[9] F. Fund, C. Wang, Y. Liu, T. Korakis, M. Zink, and S. S. Panwar, "Performance of dash and webrtc video services for mobile users," in *Packet Video Workshop (PV), 2013 20th International*. IEEE, 2013, pp. 1–8.

[10] A. Begen, T. Akgul, and M. Baugher, "Watching video over the web: Part 1: Streaming protocols," *IEEE Internet Computing*, vol. 15, no. 2, pp. 54–63, 2011.

[11] V. Swaminathan and S. Wei, "Low latency live video streaming using HTTP chunked encoding," in *2011 IEEE 13th International Workshop on Multimedia Signal Processing*. Institute of Electrical and Electronics Engineers (IEEE), Oct 2011.