International Journal of Computational and Electronic Aspects in Engineering
Volume 5, Issue 2, June 2024, pp 46 - 53
https://doi.org/10.26706/ijceae.5.2.20240506

Streaming Video Over Heterogenous Systems Using Real Time Protocol

Hesham Hashim Mohammed [*], Abdulwahhab F. Shareef, Shahla A. Abdulqader

Mosul Technical Institute, Northern Technical University, Mosul, Iraq

[*] Corresponding Author: hesham@ntu.edu.iq

Abstract: Streaming video in real time from one device to another is a challenging issue for many purposes. One of them is that both sides must initialize the connection at relatively short period of time, another issue is happened if the systems work on different platforms or different Operating Systems (OS). Network status also plays an important role in streaming quality. This paper proposes streaming live video using Real Time Protocol (RTP) form mobile camera (Android system) to a computer (Windows system). Six Streaming metrics has been taken into consideration to measure the results. These metrics are video Jitter, frame Latency, Network Throughput, Peek Signal to Noisy Ratio, and CPU usage. The results shows that network bandwidth effect on these metrics with different perspectives.

Keywords: Real Time Protocol, User Datagram Protocol, Video Streaming

Article – Peer Reviewed
Received: 30 March 2024
Accepted: 20 May 2024
Published: 7 June 2024

Cite this article: Hesham Hashim Mohammed, Abdulwahhab F. Shareef, Shahla A. Abdulqader, “Streaming Video Over Heterogenous Systems Using Real Time Protocol”, International Journal of Computational and Electronic Aspects in Engineering, RAME Publishers, vol. 5, issue 2, pp. 46-53, 2024.
https://doi.org/10.26706/ijceae.5.2.20240506

1. Introduction

Nowadays, applications which concern in social communication need to adapt a method for streaming videos between or among devices, applications like TicToc, Instagram, Facebook, and other social media applications supplied with such ability as well as to their original function as social media applications[1] [2]. Streaming video operations affect with different factors. One of them are network status at streaming time. This is due to the fact that user datagram protocol which is the protocol that responsible for delivering packets is best effort protocol, which transfer packets with no guarantee, so in case of dropping of packet losing the senders doesn’t know, on the other hand the receiver will note a denial of service which appears as low-quality video or trimmed frames[3].

Systems communication divided in generally into two categories, homogeneous and heterogeneous. Homogeneous mean that both sender and receiver of the messages working on the same platform such as Android, IOS, Windows, etc. heterogeneous communication, on the other hand mean that the message sender and receiver runs on different platforms. Obviously, the latter is more difficult to implement because using such kind of communication need to handle the internal or individual differences between the systems[4], [5].

Transmission Control Protocol TCP and User Datagram Protocol are protocols in transport layer, which is the fourth layer of TCP/IP Protocol Suite. They have a basic difference in service mechanism. This is because TCP provide transport services for the packets such as reliable in order delivery as well as packet flow control and congestion control. This made the protocol suitable for reliability nature applications, but this reliability imposes high latency in transporting operation. UDP On the other hand, doesn’t provide any reliability in the sake of low latency service [6]. If reliability is necessary then UDP must use high protocol such as RTP.

Real Time Protocol RTP is a protocol that works on the top of UDP to provide reliable transmission over the network with as low latency as possible

This paper proposes streaming video from one side to another using the services of RTP. Stream sender is a camera of mobile which runs over Android operation system, while the receiver is an application built using Python language that works on Windows operating system to receive and display such video and calculate a set of streaming metrics for it.

The rest of this paper is organized as follow. Section II offers a set of related works and the challenges encounter each one of them as well as their pros and cons. Section III offer the methodology used to stream the video over RTP. Section IV discuss the results, while section 5 discusses the related works that can extend this work.

Results measured in this paper calculated depending on 5 metrics to measure the effect of bandwidth on the streaming process. Theses metrics are: -

A. video Jitter: - the jitter in video means the ration of loosed video frames comparing with the complete frames. It's worth mentioning that as jitter increases the video reassembly become more complicated[7] [8].

B. Frame Latency: the latency means the difference in time between the instant at which the frame captured in sender device and the instant at which the frame displayed at receiver side[9] [10].

C. Throughput: throughput mean the utilization of using the network during the video streaming process[11].

D. Peek Signal to Noisy Ratio (PSNR): this metric measures the amount of noisy in the received video with respect to the original video stream[12].

E. CPU usage: this metric measures the utilization of CPU during the streaming process[13].

2. Related Works

Gatimu, et. al in 2020 propose FDTU which is a combination of TCP and UDP dependent system with name Flexible Dual TCP UDP streaming protocol. This protocol takes the benefits of both TCP protocol in terms of reliable delivery, and UDP protocol in terms of low latency. This is done by dividing the video data into two portions the most important part is delivered via TCP, while the rest of the video delivered via UDP. FDTU use the bitstream prioritization to determine the portion of video data which sent using TCP, this metric can be adjusted according network status. Comparing to TCP based video streaming, results show that the total rebuffering are enhanced 90% while packet losing decreases 70% comparing to UDP based method [6] .

To handle the security issues emerged with media end to end encryption, Yorozu, et. al propose secure RTP (SRTP) as well as Zimmermann RTP (ZRTP) protocols to handle the encryption process. The work is available as open source library named uvgRTP library. The results shows that the 10Gbps network can transfer 8K VVC video with 187 fps on Intel Core i7-4770 Micro Processor. The results also shows that 8K HEVC video with 120 fps can also be encrypted and transferred over the network[14].

In 2022, Heryana, et. al propose combining RTP with UDP Protocols to handle the latency for self driving teleoperation. In self driving teleoperation the latency can be caused by hardware, software, or network reasons in all cases it should be less than 50 milliseconds. Results shows that researchers achieves 300 ms latency which still behind the desired value[15].

In 2023 Zeng, et. al propose an approach for handle the 360 degree video stream. Streaming 360 degree video have a set of challenges one of them is that the client must merch different stream tails to regenerate the full view. To do so, researchers propose selecting and merging tile streams on Content Delivery Network (CDN) server after that the stream transmitted to the client. Researchers also adopt prefetch strategy in order to reduce disk overhead from parallel reads. Results shows that the proposed system provides slightly low latency in video transmission, low CPU usage, and good disk performance under heavy service load[16].

In 2024, Zhang, et. al[17] propose using Loki-plus to improve learning robustness via coherently integrating it with a rule-based algorithm. To perform the goal of integrity in deep features level, researchers first use an inverse re-engineering to the rules to convert it in form of black box neural network, and then devise the transform base continual learning model by emerging from effective historical feature reservation.

3. Methodology

This section describes the methodology used to perform the streaming process for live video from sender to the receiver. This section divided into two subsections the first one describes the process at sender side, while the other subsection describes the process needed at the receiver side.

A. **Sender Side:** the sender device properties are listed in table 1 below.

Table 1: sender device properties

No	***Name of Property***	***Characteristics***
1	Mobile Name	Tecno CK7n
2	Screen Resolution	1080 x 2400 pixels
3	Screen Ratio	20:9
4	Operating System	Android 13, HIOS 13
5	ChipSet	Mediatek Helio G99 (6nm)
6	CPU	Octa-core
7	Memory	8GB +8GB supplementary
8	Video	1080p with 30fps
9	Battery	5000 mAh, non-removable

The sender converts the device to an IP Camera, this is done by installing IP Webcam Pro application, which available online at google store on the following Link

https://play.google.com/store/apps/details?id=com.pas.webcam.pro&hl=en_US&pli=1.

The application is quite simple in term of use. You just need to connect it online and give it an IP address and port number then start the streaming process. The packets are sent using RTP protocol over UDP form the sender to the receiver. The sender side snapshot illustrated in figure 1 below.

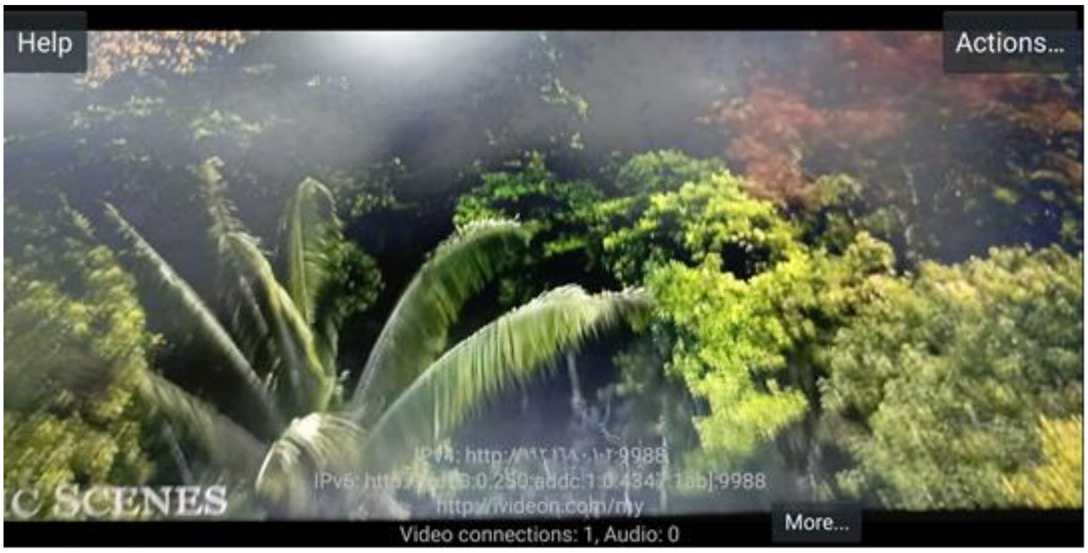
Help
Actions...
C SCENES
IPv4: http://[illegible]:9988
IPv6: http://[illegible]:0:250:addc:1:0:4347:1ab]:9988
http://ivideon.com/my
Video connections: 1, Audio: 0
More...

Figure1. Sender Side snapshot.

Form previous figure you can note that there are four liens in bottom most side of the screen, the first one represents the socket which usually written in the form of [IP: Port], the IP is of Version 4. The next line represents the socket in form of IPv6. The third line represent the URL of the program owner, and the last line represent the number of connections or receiving devises which is 1.

B. **Receiver Side:** as mentioned earlier, receiver side is a computer with Windows operating system. The receiver characteristics are listed in table 2.

Table 2: Receiver Device Properties

No	*Name of Property*	*Characteristics*
1	Device Name	HP
2	Operating System	Windows 11 Pro
3	Operating System Architecture	64-bit operating system x64-based processor
4	CPU	12th Gen Intel Core i7-1255U 1.70 GHz
5	Memory	8GB

The device supplied with Python PyCharm Community Edition 2024.1.1 it runs on top of python 3.7 interpreter. This receiving process is done as the algorithm in figure 2 illustrates.

First of all, the receiver must initialize the application one of the important things that must be set is the IP address and the port number. The IP represent the address of the destination while the port number represent the process number at the destination machine. After that the receiver still receive video frames and display them on the screen. This process terminated when the sender stops packet sending, see figure 2.

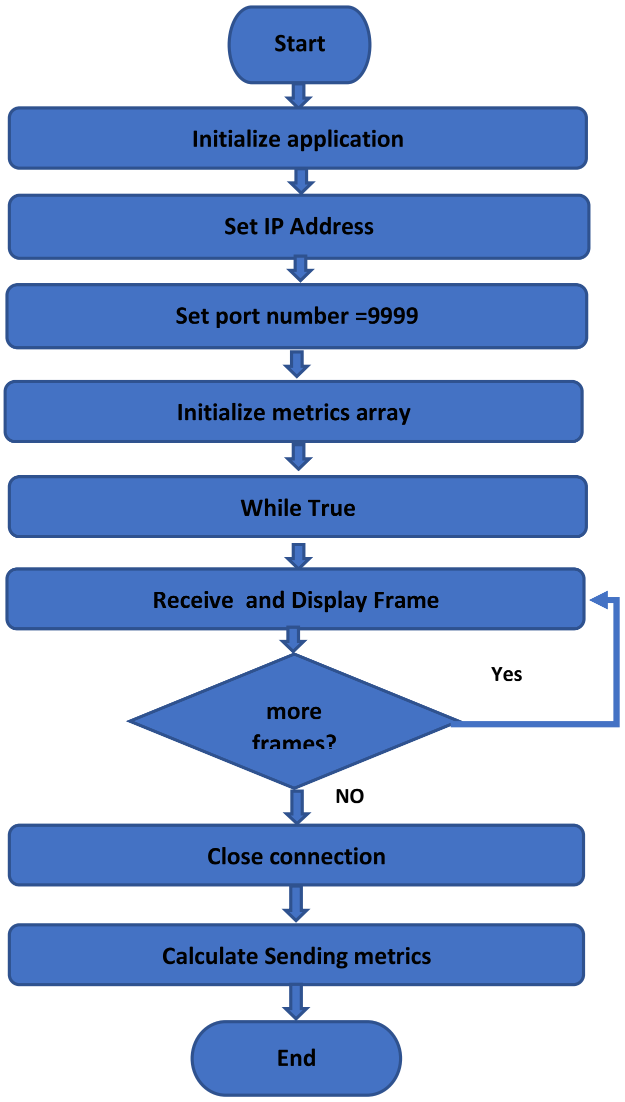

Figure 2. Receiver Side Flow Chart.

4. Result Analysis

This section will show the results of our experiments. This paper performs 5 experiments each one with particular band width where the experiments are (0.5, 1.0, 1.5, 2, 2.5) Mbps respectively. This will measure the effect of bandwidth on streaming process.

The following table illustrates the results gained from the five experiments

Table 3: Experiment Result

No	Bandwidth (Mbps)	Jitter	Latency	Network Throughput	PSNR	CPU usage
1	0.5	$85*10^{-2}$	0.0093	23	31.63	13%
2	1	$92*10^{-2}$	0.0092	42	31.61	13%
3	1.5	$98*10^{-2}$	0.0090	48	31.60	15%
4	2	$13*10^{-1}$	0.0090	51	31.60	15%
5	2.5	$15*10^{-1}$	0.0087	57	31.62	16%

As Table 3 clarify, network bandwidth significantly affects the system behavior.

You can note that the jitter, which represent the loss of frames during transmission, decreases as the bandwidth increases see figure 3.

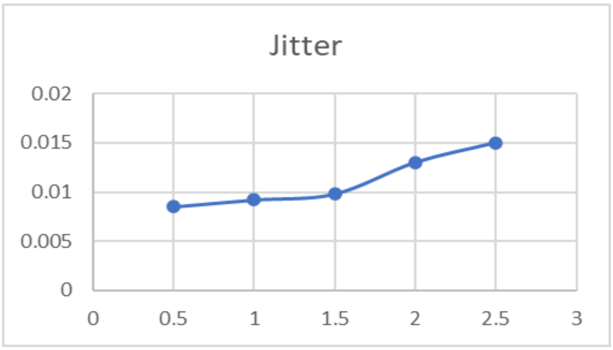

Figure 3. Streaming Video Jitter

On the other hand, the latency decreases as the bandwidth increases. This is logical because the network can transfer more data as the bandwidth increases see figure 4.

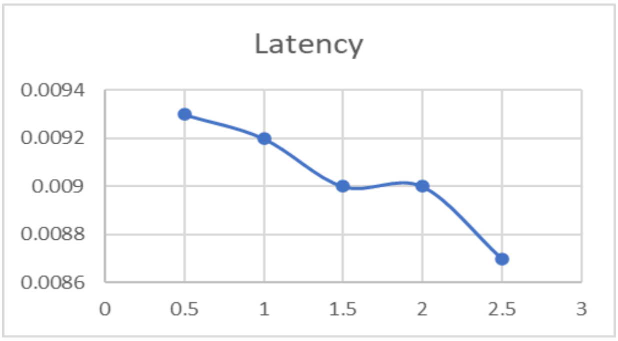

Figure 4. Streaming Frame Latency

As figure 5 illustrates Network throughput also increases because as the bandwidth increases. No significant change happened on PSNR, this is logical because PSNR measure the signal noisy See Figure 6.

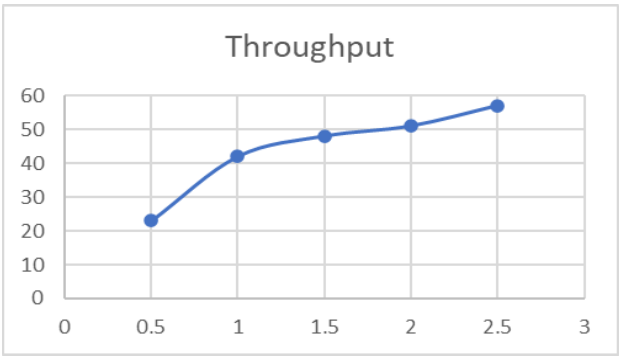

Figure 5. Throughput

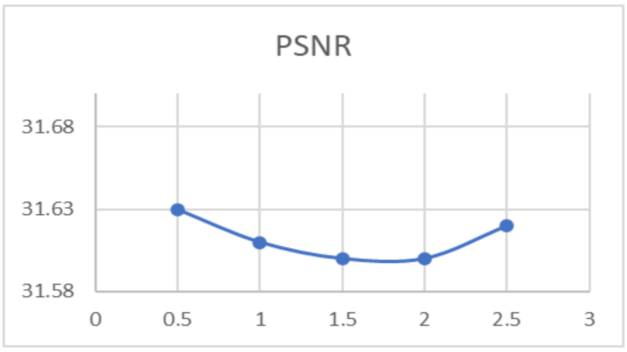

Figure 6. PSNR Value

CPU utilization also increases as the bandwidth increase. This is logical due to the need of more computation as more frames received see figure 7.

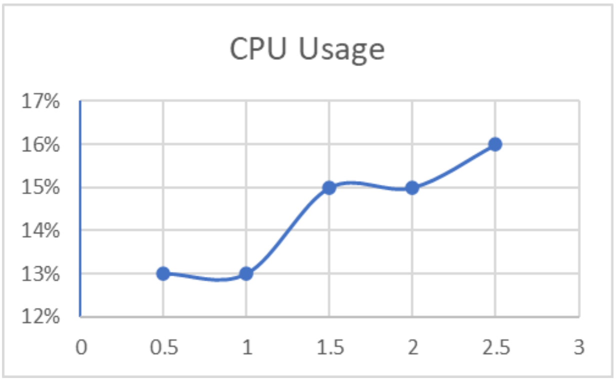

Figure 7. CPU Utilization

5. Conclosion and future works

This paper used to stream video from mobile device to PC. The difference in platforms affects on the communication process comparing with system with homogeneous platforms. Nevertheless, this affect is negligible in terms of streaming delay due to the high computation capacity. Five experiments have been performed to measure the effect of increasing bandwidth on the streaming process. Five measures have been chosen to measure the streaming process, these measures are (video Jitte, frame Latency, Network Throughput, PSNR, and CPU usage).

6. Refrences

[1] E. D. Thorburn, “Social Media, Subjectivity, and Surveillance: Moving on From Occupy, the Rise of Live Streaming Video,” *Commun. Crit. Stud.*, vol. 11, no. 1, pp. 52–63, Jan. 2014, doi: 10.1080/14791420.2013.827356.

[2] A. Castro Higueras, J. P. Pérez-Rufí, J. L. Torres Martín, M. R. Carballeda Camacho, and M. De Aguilera Moyano, “Streaming de vídeo, cómo las plataformas condicionan el comportamiento y los usos expresivos de los usuarios de sus apps,” *Rev. Lat. Comun. Soc.*, no. 80, pp. 1–20, Mar. 2022, doi: 10.4185/RLCS-2022-1545.

[3] Y. Li, C. Wang, and J. Liu, “A Systematic Review of Literature on User Behavior in Video Game Live Streaming,” *Int. J. Environ. Res. Public. Health*, vol. 17, no. 9, p. 3328, May 2020, doi: 10.3390/ijerph17093328.

[4] S. Wang, S. Yang, and C. Zhao, “SurveilEdge: Real-time Video Query based on Collaborative Cloud-Edge Deep Learning,” in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, Toronto, ON, Canada: IEEE, Jul. 2020, pp. 2519–2528. doi: 10.1109/INFOCOM41043.2020.9155284.

[5] W. Liang, G. Wang, J. Lai, and X. Xie, “Homogeneous-to-Heterogeneous: Unsupervised Learning for RGB-Infrared Person Re-Identification,” *IEEE Trans. Image Process.*, vol. 30, pp. 6392–6407, 2021, doi: 10.1109/TIP.2021.3092578.

[6] K. Gatimu, A. Dhamodaran, T. Johnson, and B. Lee, “Experimental study of QoE improvements towards adaptive HD video streaming using flexible dual TCP-UDP streaming protocol,” *Multimed. Syst.*, vol. 26, no. 4, pp. 479–493, Aug. 2020, doi: 10.1007/s00530-020-00653-w.

[7] C. Ghasemi, H. Yousefi, and B. Zhang, “Internet-Scale Video Streaming over NDN,” *IEEE Netw.*, vol. 35, no. 5, pp. 174–180, Sep. 2021, doi: 10.1109/MNET.121.1900574.

[8] Y. Urgun and A. Kavak, “SDN/NFV-based QoS mechanism for Cellular Core Networks,” in *2016 24th Telecommunications Forum (TELFOR)*, Belgrade, Serbia: IEEE, Nov. 2016, pp. 1–4. doi: 10.1109/TELFOR.2016.7818733.

[9] S. Wei and V. Swaminathan, “Low Latency Live Video Streaming over HTTP 2.0,” in *Proceedings of Network and Operating System Support on Digital Audio and Video Workshop*, Singapore Singapore: ACM, Mar. 2014, pp. 37–42. doi: 10.1145/2597176.2578277.

[10] M. A. Usman, M. R. Usman, and Soo Young Shin, “A no reference method for detection of dropped video frames in live video streaming,” in *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*, Vienna, Austria: IEEE, Jul. 2016, pp. 839–844. doi: 10.1109/ICUFN.2016.7537155.

[11] X. Zhang and H. Hassanein, “Video on-demand streaming on the Internet &#x2014; A survey,” in *2010 25th Biennial Symposium on Communications*, Kingston, ON, Canada: IEEE, 2010, pp. 88–91. doi: 10.1109/BSC.2010.5472998.

[12] M. Martini, “A Simple Relationship Between SSIM and PSNR for DCT-Based Compressed Images and Video: SSIM as Content-Aware PSNR,” in *2023 IEEE 25th International Workshop on Multimedia Signal Processing (MMSP)*, Poitiers, France: IEEE, Sep. 2023, pp. 1–5. doi: 10.1109/MMSP59012.2023.10337706.

[13] Z. Fu, J. Zhou, W. Xu, C. Guo, and Q. Wu, “GPU and VPU Enabled Virtual Mobile Infrastructure for 3-D Image Rendering and its Application in Telemedicine,” *IEEE Internet Things J.*, vol. 11, no. 5, pp. 7724–7738, Mar. 2024, doi: 10.1109/JIOT.2023.3316698.

[14] J. Rasanen, A. Altonen, A. Mercat, and J. Vanne, “Open-source RTP Library for End-to-End Encrypted Real-Time Video Streaming Applications,” in *2021 IEEE International Symposium on Multimedia (ISM)*, Naple, Italy: IEEE, Nov. 2021, pp. 92–96. doi: 10.1109/ISM52913.2021.00023.

[15] A. Heryana *et al.*, “Realtime Video Latency Reduction for Autonomous Vehicle Teleoperation Using RTMP Over UDP Protocols,” in *Proceedings of the 2022 International Conference on Computer, Control, Informatics and Its Applications*, Virtual Event Indonesia: ACM, Nov. 2022, pp. 44–49. doi: 10.1145/3575882.3575891.

[16] Q. Zeng *et al.*, “TVSR-OR: Tile-based 360-degree video streaming over real time streaming protocol with optimized read,” *Trans. Emerg. Telecommun. Technol.*, vol. 34, no. 5, p. e4759, May 2023, doi: 10.1002/ett.4759.

[17] H. Zhang, A. Zhou, G. Wang, C. Li, and H. Ma, “Toward Optimal Live Video Streaming QoE: A Deep Feature-Fusion Approach,” *IEEEACM Trans. Netw.*, pp. 1–16, 2024, doi: 10.1109/TNET.2024.3351832.