A STUDY OF ISLAND NETWORK PERFORMANCE FOR STREAMING PROTOCOLS

Ammar Hatem Mohammed

A thesis submitted in
fulfillment of the requirement for the award of the
Degree of Master of Electrical Engineering

Faculty of Electrical and Electronic Engineering
Universiti Tun Hussein Onn Malaysia

JUNE 2019

For my beloved children Mohaimen & Rima

ACKNOWLEDGEMENT

Praise be to God the Almighty for everything, I believe all the successes I obtained in my life are not by my own powers but they are graces from Allah. I would like to express my deep thanks and gratitude to my supervisor: Assoc. Prof. Ir. Dr. Abd Kadir Bin Mahamad, for his valuable guidance and support given throughout the duration for this research.

Besides my supervisor I want to thank my university. University Tun Hussein Onn Malaysia for giving me this scholarship, and I do not forget my professors at the Department of Computer Engineering, Faculty of Electrical and Electronic Engineering UTHM.

In addition, I want to thank my 1st teacher “Mother” without her prayers for me I wouldn't have gotten what I have reached today, also big thanks for my Father, brothers, my beautiful wife and my beloved children for help and support to complete my studies and get my dream, and I thank everyone who helped me complete my studies and complete my research.

'Praise be to God'

ABSTRACT

Nowadays video surveillance is a cornerstone of the security in the world. It provides real-time monitoring for alarm of the environment, for people as face recognition, for property as plate car numbers detection, and provides a recorded archive for investigation. With megapixel cameras becoming increasingly widespread, even the bandwidth exhaustion of corporate networks is becoming a real issue. In this research, study on the performance of the island network using streaming protocol of HTTP and RTSP to broadcast the IP camera when streaming executed on H264 and H.265 encoder was conducted. The research done on the real island network that build to use as test bed for the project, also used network emulator (NetEm) to inject the packet loss and delay to the island network to emulate real big network. Then the results were analysed by Wireshark packet analyser. Based on the results gained, it was found that HTTP over TCP has less packets when compared to RTSP. As a conclusion, Hypertext Transfer Protocol is a little superior and authoritative protocol to stream a video when compared to the RTSP protocol.

ABSTRAK

Pengawasan video pada masa kini merupakan asas keselamatan di dunia. Ia menyediakan pemantauan masa nyata untuk penggera alam sekitar, untuk orang sebagai pengesanan muka, untuk harta benda sebagai pengesanan nombor plat kereta dan menyediakan arkib yang direkodkan untuk penyiasatan. Dengan penggunaan kamera megapiksel yang semakin meluas, ditambah pula dengan ketidakcekapan jalur lebar rangkaian korporat menjadikan ianya isu yang sebenar. Dalam kajian ini, kajian mengenai prestasi rangkaian pulau menggunakan protokol penstriman HTTP dan RTSP untuk menyiarkan perlaksanaan kamera IP semasa penstriman pada pengekod H264 dan H.265. Penyelidikan ini dijalankan di rangkaian pulau sebenar yang dibina untuk digunakan sebagai tempat ujian bagi projek ini, yang menggunakan emulator rangkaian (NetEm) untuk menyuntik kehilangan paket dan kelewatan ke rangkaian pulau untuk menyamai rangkaian besar yang sebenar. Kemudian keputusan dianalisis oleh penganalisis paket Wireshark. Berdasarkan hasil yang diperoleh, didapati HTTP melalui TCP mempunyai paket yang kurang jika dibandingkan dengan RTSP. Sebagai kesimpulan, Protokol Pemindahan Hypertext adalah protokol yang lebih unggul dan berwibawa untuk menstrimkan video jika dibandingkan dengan protokol RTSP.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

AVC	-	Advanced Video Coding
CCTV	-	Closed-circuit television
CVS	-	Cloud Video Storage
DASH	-	Dynamic Adaptive Streaming over HTTP
FPS	-	Frames per second
GUI	-	Graphical User Interface
HAS	-	Hypertext transfer protocol Live Streaming from Apple
HEVC	-	High Efficiency Video Coding
HTTP	-	Hypertext Transfer Protocol
ICMP	-	Internet Control Message Protocol
IP	-	Internet Protocol
IPC	-	IP Camera
JPEG	-	Joint Photographic Experts Group
Kbps	-	Kilobit per second
LAN	-	Local area network
Mbps	-	Megabit per second
MPEG-4	-	Moving Picture Experts Group
NEWT	-	Network Emulator for Windows Toolkit
NVR	-	Network video recorder
ONVIF	-	Open Network Video Interface Forum
QoE	-	Quality of Experience
QoS	-	Quality of Service
RTCP	-	RTP Control Protocol
RTMP	-	Real-Time Messaging Protocol
RTP	-	Real-time Transport Protocol
RTSP	-	Real Time Streaming Protocol
RTT	-	Round-Trip times
TCP	-	Transmission Control Protocol
UDP	-	User Datagram Protocol
VLC	-	VideoLan Client
VPN	-	Virtual Private Network
WAN	-	Wide area network

CHAPTER 1

INTRODUCTION

In recent years, there has been an increase in the number of IP cameras especially for medium and big CCTV project rather than analog cameras which the live streaming services are also demanded. Streaming high quality live camera has become a challenge for the enterprise network infrastructures. There are several protocols for streaming the video. The network engineers are mainly focused not only on the perceived quality of live IP camera for the observers but also for the recording in central video storage. There has been a huge development of applications for the minimization of the errors to enhance the observer's experience. Choosing an appropriate streaming protocol is a challenging task for the manufacturers. The performance of the protocols depends on the network parameters such as packet loss, delay and available capacity. This indicates that observers can view the same live cameras at the same time over the network.

There are different protocols in network that are used for streaming the IP camera. There has been a great development and research in video streaming protocols field like HTTP (Hyper Text Transfer Protocol), RTSP (Real-Time-Streaming-Protocol), RTP (Real-Time-Transport-Protocol), and RTMP (Real-Time-Messaging-Protocol).

In this first chapter is discussed the background of the video streaming principles. In addition, the objectives of project and the project scopes are also discussed in deep.

1.1 Background of the project

Streaming live video become very useful way to monitor the live for camera directly on video wall, which the selection of transport protocols based on UDP or TCP makes an impact on streaming performance. Therefore, in this project, have been discussed some standard of video streaming protocols.

HTTP (HyperText Transfer Protocol) has grown a lot in terms of its usage for live streaming, whereas RTSP (Real Time Streaming Protocol) has been familiar in the market for a long time and also a good alternative for streaming videos. HTTP has incalculable implementations on both server and client, whereas RTSP is used for controlling media session between end points. HTTP protocol doesn't require additional special proxies or caches. It is a stateless protocol and behaves as a system without a feedback control for multimedia transmission [1].

When an HTTP client requests the data, the server responds to the request by delivering the requested data. Each HTTP request is handled totally independently [2]. HTTP media streaming is the cheapest and the easiest to deploy as it does not require specialized servers at network nodes. It is relatively easy to traverse middle boxes such as firewalls and Network Address Translation devices, keeping up the essential information on the server side, which makes HTTP servers more scalable than normal streaming servers.

RTSP is a protocol used for control media over the delivery of real time data in an application level to address the needs for efficient delivery of the data over the internet. It is specifically designed to minimize the overhead of multimedia transmission over IP network [3]. This protocol is designed to control multiple data sessions and provide means for choosing different delivery channels such as UDP, multicast UDP and TCP [4]. RTSP works on top of a well-established Real Time Transport Protocol (RTP), which RTSP establishes a TCP connection between the server and the client. When a client sends out controlling requests such as PLAY, PAUSE, etc., to the server, it achieves real time control of playback of the media files from the servers. RTSP is a stateful protocol and it tracks the data and records the media transitions [1][3].

The High Efficiency Video Coding (H.265/HEVC) is a newly standardized encoder that saves up to 50% of the bandwidth when compared to Advanced Video Coding (H.264/AVC) when streaming the video content over wireless networks [5][6].

The main improvement made in this encoder is due to it can support high resolution video. Metrics like delay, bandwidth and packet loss, etc. are important for measuring the performance of the video streaming. Among these, packet loss and packet delay variation are the important factors that are used for calculating the performance of the video stream as it may result in degradation of the quality of video [4] .

1.2 Problem statement

Surveillance cameras had become a pressing necessity in our daily lives in safe city projects, companies, banks, houses and almost everywhere. Because of the increasing number of surveillance cameras and resolution for each camera it's become in Megapixels. In addition, it became necessary to find the best way to streaming the IP cameras for many monitoring center with the lowest bandwidth as possible with the best quality and minimum delay.

It is important that surveillance cameras streaming data is not being on the internet network for many reasons such as security information, that it should be in an island network.

There is always a limit number of the surveillance IP cameras to remote access, when there are many users wanted to get remote access to same IP cameras to get the live view, or to recoding in different devices such as NVRs, CVS and etc. Thus, in this research it evaluates the performance of streaming protocols in the island network when using IP camera on encoder standards to lower the load on the network devices.

1.3 Objectives of the project

The aim of the project is to evaluate the performance of the island network when using IP cameras on encoder standards. The performance at network layer have been analysed for different streaming protocols in this project. The obtained results are then have been analysed. The objectives that are implemented to achieve the goal are mentioned below:

1. Design and build an island network in order to identify performance of the experiment in real network with IP cameras.
2. Propose an appropriate streaming protocols, IP cameras, stream media server and encoder standards to identify the relevant features to the research.
3. Evaluate the performance analysis of the propose island network on each protocol.

1.4 Scope of the project

This thesis mainly describes about the network evaluation of streaming protocols when using IP cameras on encoder standards. HTTP and RTSP protocols are examined in this project and metrics like packet loss, delay/latency and bitrate are considered in order to evaluate the performance of these protocols. The performance is also evaluated when variable delay and bandwidth restrictions are injected into the network.

The Cisco Island network was setup to perform the research work. IP cameras has been used in order to capture the live streaming; VLC Media Player is used to evaluate the performance of different protocols. Also, Wireshark which is an open-source packet analyzer has been used in order to calculate the data from the island network. Network Emulator for Windows Toolkit - NEWT has been used for injecting delay, packet loss and restricting bandwidths into the network.

To implement the above-mentioned approach, VLC Media player and IP cameras have been configured on the Windows Operating System. The results obtained from the above approaches were analysed and compared.

1.5 The project report outlines

The outlines of the project reported as the follows:

Chapter 1 introduces the project objective, project scopes and problem statements, while Chapter 2 presents a survey about the major issues in island network and Streaming Protocol as well as an overview of the important part related to the project. The methodology of the project, parameters and tools, which has been used to implement in this project has been explained in Chapter 3. The results of the project in order to evaluate the performance of island network based on different streaming protocol is discussed in Chapter 4. To sum up, the project conclusion and the future works has been discussed in chapter 5.

CHAPTER 2

LITERATURE REVIEW

This chapter summarizes the information of the related previous studies on streaming protocol. In addition, this chapter covers types of live streaming protocol, the related video encoder standards and the experiment environment. These reviews are done based on materials from journals, conference proceedings and books.

2.1 History overview of video streaming

When the 1st commercial streaming services appeared in late 1992, the networks had comparatively huge Round-Trip times (RTT) [7] and provided low data rates than what user can uses today. These limitations significantly impacted the design choices relating to transport protocols.

At that time most of streaming video services was intended by using UDP, instead of Transmission Control Protocol (TCP). The choice to use UDP was typically motivated by the ability of the sender to transmit at the playback rate. However, UDP offers no guarantees of packet delivery, ordering, or error-correction. In contrast to UDP, Transmission Control Protocol present reliable connection-oriented services for congestions, flow-control, but doesn't allow the sender to control the send rate, since the rate at which packets are delivered depend on the congestion and flow control mechanisms. While UDP can provide timely delivery of packets, the streaming services had to add functionality to overcome missing and out-of-order packets. To allow the streaming player to recover for same situations without more retransmissions, different error control and concealment techniques [7], forward error correction , and other schemes to mask imperfections were used. Such techniques were typically used with early streaming protocols such as (MMS), Real-Time-Messaging-Protocol (RTMP) [8] , and the (RTP) and its related suite (Real Time Control Protocol (RTCP) and (RTSP) [3]) to mention a few.

2.1.1 Video streaming over the web

The use of proprietary or UDP-based protocols had major limitations for streaming. Since the most important among these was that traffic over outwardly initiated UDP or non-standardized protocols are usually blocked by Network Address Translators, firewalls, because of security issues with connectionless services using UDP and the risks posed by anonymous protocols. On the other hand, many protocols at the time required the servers to track the client state continually, requiring dedicated infrastructure, and intelligence at the server side too. These restrictions together with faster internet speeds (that allowed client buffers to be filled quickly), slowly began to outweigh the advantages of using User Datagram Protocol -like protocols. While, the World Wide Web has grown tremendously in the last decade. The WWW uses the Hypertext Transfer Protocol (HTTP) as its application layer protocol and the TCP protocol to ensure that all bytes of a webpage are (eventually) delivered. The development of content delivery networks CDNs and proxy caches remarkably reduced the cost incurred by the content provider and the network operator in delivering data to end users. Incremental improvements to access speeds [9], and round-trip times [4] over the years have made streaming over TCP a possible alternative. In addition, as client-side computational power and storage capacity is better, the clients could use a huge buffer to accommodate for short-term fluctuations in the network bandwidth. The comparatively huge buffer also provides additional time for TCP’s error correction and recovery protocols to recover packets in time for playback [10].

2.2 Related Work

Video stream needs a real time performance in network to deliver the video content, typically, RTP that is employed over the IP network [11]. The video supply was an IP camera.

The code to try is using video data formatting that is accessible from many corporations, which every corporations has its proprietary ways. A number of the acquainted names for example; Windows Media Player from Microsoft; VLC media player by VideoLan; QuickTime from Apple; and Adobe. Newer video-streaming standards like hypertext transfer protocol Live Streaming (HAS) from Apple are developed to support video streaming to iPhones and different sensible mobile devices. These normal uses hypertext transfer protocol (Hypertext Transfer Protocol) IP technology as against RTSP, which might enable to bypass several firewalls in IP networks. Microsoft offers smooth streaming, that additionally dispenses with RTSP in favour of hypertext transfer protocol IP technology.

HTTP invented there isn't planning to make it working for video streaming, however it's been found to be very powerful. Recently, a keep – a live mechanism was introduced, which an association may well be reused for quite one request. By employing a persistent connection, it can reduce the latency, as a result of the transmission control protocol connection doesn't have to be compelled to be re- negotiate once the primary request has been sent.

Most of the discussion to the present purpose has been concerning unicasting, or causing video from one supply to at least one destination. However, several applications are multicast, like live events football games, broadcast IPTV or security system IPC and traffic cameras.

In [12] they compare 3 completely different protocols hypertext transfer protocol, RTSP and Dynamic Adaptive Streaming over HTTP (DASH) on smartphones. They need calculated switch delay, that is the delay amount between the user causing a switch command and the consumer screen begins to point out frames of latest perspective at the normal video quality delineated. However, the screen acts once the consumer stops showing the recent perspective and before it shows the new perspective and restricted the information measure. They solely used H.264 encoder and enforced within the media player of the smartphone. The authors examined Dynamic Adaptive Streaming over HTTP (DASH) based, which mostly gave best performance because it provided seamless shift with a delay of at the most two seconds whereas solely acquisition 100% overhead.

Authors in [13] have conducted associate experiment victimisation network machine on hypertext transfer protocol, mostly video transmission and analysis on the network impairment influence the streamed video. Authors confirmed the experiment against a network that equipped real network traces. Finally, the authors expressed that the buffering methods enforced by a video player are in several cases is able to mitigate unfavourable network conditions that permits the streamed video to play swimmingly. Authors in [14] have given a QoE instrumentation for video streaming, VLQoE on a smartphone. They need supplemental of practicality to the VLC media player to record a group of metrics from the programme, application-level, network-level and from the obtainable sensors of the device. A tool has been won't to gift a 2-state model supported the inter-picture time for the hypertext transfer protocol and RTSP based mostly video streaming via 3.5G. Author study has been done on the influence of inter-picture time on the user's perceived quality. Finally, the authors have all over the lay image times varies from user to user (max 2880ms, min 40ms). The authors have additionally expressed that most image time increased because the user rating decreased and also the highest mean of maximum inter-picture times have matched with the "Freeze" indications. They need additionally analysed the user latency of the themes in 2 situations like short and long freezes. They need found that the user latency for long freezes was exponentially distributed.

In paper [15], the authors have explained the technical and non-technical parameters of QoE. Within the technical parameters, the authors have thought of network level and application level QoS and for the non-technical the authors have thought of perception, expertise and expectations.

Always these communication classes are split into Push-Based and Pull-Based protocols [16]. Push-based protocols comprised established communication from the client to the server, wherever the responsible for establishing the connection is the client, and also the server sends a packet stream till the client stops or interrupts the communication. During this form of protocol, the server, additionally to caution media packets, maintains a session to pay attention for commands from the client. These protocols always working with (UDP), sometimes with (TCP) but commonly use (RTP).

Pull-based protocols of the streaming depend on the bandwidth on the network due to it's based on HyperText Transfer Protocol (HTTP), thus the client sending a request to the server, which the server starts a communication where the client downloads the video streaming, such as progressive download.

Table 1.1 shows some examples of streaming protocols and their features [17], which an analysis of the transmission protocols for the H.265 encoder.

Table 1.1: The streaming protocols Comparison Push based and Pull based

Feature	Push-Based	Pull-Based
Source	Broadcasters and servers like Windows media, QuickTime, Real Networks Helix Cisco CDS/DCM	Web servers such as LAMP, Real Networks Helix, Microsoft BS, Cisco CDS
Protocols	RTSP, RTP, UDP	HTTP (HLS, MPEG-DASH, Adobe HTTP Dynamic Streaming, Microsoft Smooth Streaming)
Bandwidth usage	Likely more efficient	Likely less efficient
Video monitoring	RTP Control Protocol (RTCP)	Currently proprietary
Multicast support	Yes	No

2.3 IP video surveillance

There are many functions effects must to consider when planning to design an IP Video Surveillance system operate over a LAN network.

2.3.1 IPC live streaming

Live streaming is a process wherein a video is being transmitted from a camera to a destination/ multiple destination. In this thesis, a video is streamed from cameras to streaming media server then to the clients and storages devices.

Video streaming is essentially based on two main activities:

a) Creating digital content using compression methods.

b) Transmitting the content over a network.

Compression techniques are used to avoid transmitting a raw video over the network as it is burdensome. Transmitting a raw video consumes more network and storage resources. Hence, compressing a video plays an effective role in IPC streaming

2.3.2 Streaming over HTTP

With HTTP, clients typically connect to port 80 on a server. It's established a TCP connection, then the clients request data from the server using standard HTTP methods such as GET (used to retrieve content) and POST (to upload data to the server).

2.3.3 Video compression

By using video compression techniques the data of the video can be reduce and eliminate with one of the compression methods, a huge decrease in file size can be achieved with very little or no negative impact on image quality [18]. However, video quality can be affected if the file size is greatly reduced by upraise the level of compression for a particular compression technology.

There are many compression techniques, most network video companies today they use standard compression technologies. Standards are important for compatibility and interoperability. It is particularly relevant to video compression wherever video may be used for various functions, and in some video surveillance applications, it must be able for several years from the recording data. By deploying standards, end users square measure in a position to select from and for totally different suppliers, rather than connecting them to 1 supplier when planning to install a CCTV.

There are four standards for video compression. They are Motion JPEG, MPEG-4 Part 2 (MPEG-4), H.264 and H.265/HEVC the latest efficient video compression standard [19].

In the video compression, raw video is compressed using disparate algorithms and mechanisms. These are done to reduce the video and file size, which in turn reduces the consumption of network resources while in transmission. In wired as well as wireless networks, any video which is uncompressed consumes more bandwidth and storage when compared to a compressed video. In order to eradicate this problem, video compression is a must.

Video compression can be achieved in two methods:

- Lossless Compression - In this method, the original video is compressed to a smaller size wherein there isn't any loss of data. This method is not recommended for large sized videos, as it requires more network resources.
- Lossy Compression - In this method, the original video is compressed to much smaller size, which results in loss of information. This reduces the quality of the video to a great extent.

The main attributes involved in video compression are to maintain balance in between video quality and the size. Various codecs have been developed to serve this purpose.

2.3.4 Video codecs

Video compression includes an algorithm applied to the source video to create a compressed file that is ready for transmission or storage. There are two components include in compression of video are Encoder and Decoder [20], together called as CODEC, as shown in the Figure 2.1. A codec has the capability of encoding and decoding. The raw video is compressed into small sized video that is transmittable over a network. Some of the commonly used video codecs are H.265/HEVC, H.264/AVC, H.263, H.262, etc. Due to the increase in the communication of the videos, many video coding techniques have been developed in each CCTV companies to provide high quality video streams to use with the low bandwidth [21].

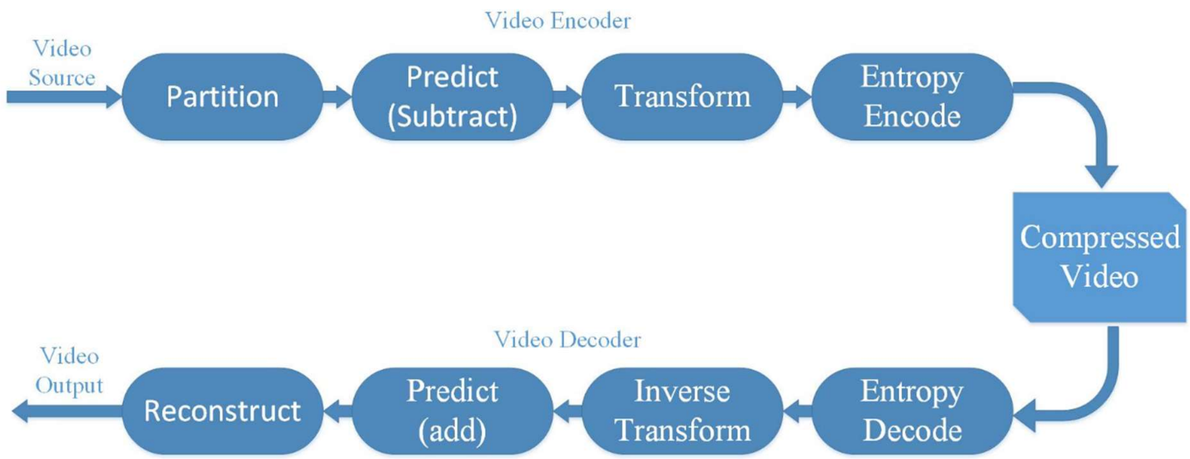

Figure 2.1: Video Codec for encoding and decoding process

2.3.5 Frame rates

It's the number of captures pictures per second called FPS, if the frame rate increase from 1 FPS to 10 FPS, the bandwidth will increase also but for the new modern codecs the increasing will not be liner such in H.264 the increasing in bandwidth usually only 3 to 5 times more.

So, the frame rate should select to meet the scene requirements, it doesn't need to be higher than what is required because it will effect on the both bandwidth and storage requirements. 24 FPS is enough to captured the motion pictures smooth due to human eye sees captured at 24 frame per second as smooth motion [22].

2.4 IP video transmission

The streaming protocols are designed to prepare data transmission, network addressing and services between main server and the clients. The transport protocols are used as a communication medium between the streaming servers and clients. TCP and UDP video

REFERENCES

1. Follansbee, J. *Get streaming!: Quick steps to delivering audio and video online*. (Focal Press, 2012).
2. Zambelli, A. IIS smooth streaming technical overview. *Microsoft Corp.* **3**, 40 (2009).
3. Wu, D., Hou, Y. T., Zhu, W., Zhang, Y.-Q. & Peha, J. M. Streaming video over the Internet: approaches and directions. *IEEE Trans. circuits Syst. video Technol.* **11**, 282–300 (2001).
4. Cohen, R. A. & Radha, H. Streaming Video. *Wiley Encycl. Telecommun.* (2003).
5. Shah, R. A., Asghar, M. N., Abdullah, S., Fleury, M. & Gohar, N. Effectiveness of crypto-transcoding for H. 264/AVC and HEVC video bit-streams. *Multimed. Tools Appl.* 1–30 (2019).
6. Adeyemi-Ejeye, A. O., Alreshoodi, M., Al-Jobouri, L. & Fleury, M. Prospects for live higher resolution video streaming to mobile devices: achievable quality across wireless links. *J. Real-Time Image Process.* **16**, 127–141 (2019).
7. Shenoy, P. J. Multimedia systems research: The first twenty years and lessons for the next twenty. *TOMCCAP* **9**, 31–38 (2013).
8. Metzger, F., Rafetseder, A. & Tutschku, K. A performance evaluation framework for video streaming. in *2012 19th International Packet Video Workshop (PV)* 19–24 (IEEE, 2012).
9. Fayazbakhsh, S. K. *et al.* Less pain, most of the gain: Incrementally deployable icn. in *ACM SIGCOMM Computer Communication Review* **43**, 147–158 (ACM, 2013).
10. Feamster, N. & Balakrishnan, H. Packet loss recovery for streaming video. in *12th International Packet Video Workshop* 9–16 (PA: Pittsburgh, 2002).
11. Tao, S., Apostolopoulos, J. & Guérin, R. Real-time monitoring of video quality in IP networks. *IEEE/ACM Trans. Netw.* **16**, 1052–1065 (2008).
12. Zhang, H., Al-Nuaimi, A., Gu, X., Fahrmair, M. & Ishibashi, R. Seamless and efficient stream switching of multi-perspective videos. in *2012 19th International*

Packet Video Workshop (PV) 31–36 (IEEE, 2012).

13. Biernacki, A. & Tutschku, K. Performance of HTTP video streaming under different network conditions. *Multimed. Tools Appl.* **72**, 1143–1166 (2014).
14. Ickin, S. *et al.* VLQoE: Video QoE instrumentation on the smartphone. *Multimed. Tools Appl.* **74**, 381–411 (2015).
15. Hoßfeld, T., Fiedler, M. & Zinner, T. The QoE provisioning-delivery-hysteresis and its importance for service provisioning in the future internet. in *2011 7th EURO-NGI Conference on Next Generation Internet Networks* 1–8 (IEEE, 2011).
16. Wingerath, W., Ritter, N. & Gessert, F. *Real-Time & Stream Data Management: Push-Based Data in Research & Practice*. (Springer, 2019).
17. UMESH, A. Performance analysis of transmission protocols for H. 265 encoder. (2015).
18. Naderiparizi, S., Hessar, M., Talla, V., Gollakota, S. & Smith, J. R. Towards battery-free {HD} video streaming. in *15th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 18)* 233–247 (2018).
19. Rajalakshmi, K. & Mahesh, D. K. An Innovative Video data Embedding Method for Video Security. *Int. J. Pure Appl. Math.* **118**, 207–213 (2018).
20. Flierl, M. & Girod, B. Multiview video compression. *IEEE Signal Process. Mag.* **24**, 66–76 (2007).
21. Richardson, I. E. *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia*. (John Wiley & Sons, 2004).
22. Watson, A. B. High frame rates and human vision: A view through the window of visibility. *SMPTE Motion Imaging J.* **122**, 18–32 (2013).
23. Stevens, W. R. *TCP/IP illustrated vol. I: the protocols*. (Pearson Education India, 1994).
24. Akoma, C. O., Igbeta, O. D. & Nlemadim, O. Intelligent video surveillance system. *Case study, Fac. Comput. London Metrop. Univ. London, UK* (2015).
25. Kogane, H., Nakamura, M. & Higashimura, M. Network surveillance video camera system. (2006).
26. Vuppala, A. & Lakshminarayana, S. Measurement of user-related performance problems of live video streaming in the user interface. (2012).

27. Joel W. King. IP Video Surveillance Design Guide - Cisco. Available at: https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Video/IPVS/IPVS_DG/IPVS-DesignGuide.html. (Accessed: 31st May 2019)
28. Froom, R., Sivasubramanian, B. & Frahim, E. *Implementing Cisco IP Switched Networks (SWITCH) Foundation Learning Guide: Foundation learning for SWITCH 642-813*. (Cisco press, 2010).
29. Shin, J. Y., Kwon, Y. S., Kim, H. H., Kim, K. D. & Shim, M. J. Method and device for media streaming between server and client using rtp/rtsp standard protocol. (2018).
30. Wireshark Developer's Guide. Available at: https://www.wireshark.org/docs/wsdg_html_chunked/. (Accessed: 31st May 2019)
31. Network emulator for windows toolkit in Microsoft visual studio. Available at: https://visualstudio.microsoft.com/. (Accessed: 31st May 2019).