Evolution of Information, Communication and Computing Systems (EICCS)
https://publisher.uthm.edu.my/bookseries/index.php/eiccs

A Study of Island Network Performance for Streaming Protocols

Ammar Hatem Mohammed[1*], Abd Kadir Mahamad[2], and Sharifah Saon[2]

[1]Independent Researcher, 40th Street, Hillah, 00964 Babylon, IRAQ

[2]Faculty of Electrical and Electronic Engineering,
Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, MALAYSIA

*Corresponding Author

Abstract:
Nowadays, video surveillance is a cornerstone of security in the world. It provides real-time monitoring for the alarm of the environment, for people as face recognition, for the 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 paper, a study on the island network's performance using a streaming protocol of HTTP and RTSP to broadcast the IP camera when streaming executed on H264 and H.265 encoder was conducted. The research is done on the real island network that has been built as a test bed for the project. A network emulator (NetEm) was also used to inject the packet loss and delay to the island network in order to emulate a real network. Then the results were analyzed by Wireshark packet analyzer. Based on the results gained, it was found that HTTP over TCP has less packets when compared to RTSP. In conclusion, Hypertext Transfer Protocol is a little superior and authoritative protocol to stream a video compared to the RTSP protocol.

Keywords:
HTTP · RTSP · TCP · IP camera · Video surveillance

1. Introduction

In recent years, there has been an increase in the number of internet protocol (IP) cameras, especially for medium and big CCTV projects rather than analog cameras. Live streaming services are also demanded. Streaming high-quality live cameras have become a challenge for the enterprise network infrastructures. The network engineers are mainly focused on the perceived quality of live IP camera for the observers and the recording in central video storage. There has been a massive development of applications to minimize errors to enhance the observer's experience. Choosing an appropriate streaming protocol is a challenging task for 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 simultaneously over the network.

There are different protocols in the network that are used for streaming the IP camera. Therefore, there has been great development and research in video streaming protocols fields like Hyper Text

*Corresponding author: czarfl55@gmail.com

Transfer Protocol (HTTP), Real-Time-Streaming-Protocol (RTSP), Real-Time-Transport-Protocol (RTP), and Real-Time-Messaging-Protocol (RTMP).

Streaming live video becomes a very useful way to monitor the live for camera directly on the video wall. However, the selection of transport protocols based on User Datagram Protocol (UDP) or Transmission Control Protocol (TCP) impacts streaming performance. Therefore, in this project, have been discussed some standards of video streaming protocols.

HTTP has grown a lot in terms of its usage for live streaming, whereas RTSP has been familiar in the market for a long time and is also a good alternative for streaming videos. HTTP has incalculable implementations on both server and client, whereas RTSP controls media sessions between endpoints. 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 by delivering the requested data. Each HTTP request is handled 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 middleboxes such as firewalls and Network Address Translation devices, keeping up the essential information on the server-side, making HTTP servers more scalable than normal streaming servers.

RTSP is a protocol used to control media over the delivery of real-time data at an application level to address the need for efficient data delivery over the internet. It is specifically designed to minimize the overhead of multimedia transmission over IP networks [3]. Furthermore, 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 RTP, in 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 that it can support high-resolution video. Metrics like delay, bandwidth and packet loss, etc. are essential for measuring the video streaming performance. Among these, packet loss and packet delay variation are the essential factors that are used for calculating the performance of the video stream as it may result in degradation of the quality of video [4].

Surveillance cameras have 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 stream the IP cameras for many monitoring centres with the lowest bandwidth possible with the best quality and minimum delay. It is important that surveillance cameras streaming data are 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 limited number of surveillance IP cameras to remote access. Many users wanted to get remote access to the same IP cameras to get the live view or recoding in different devices such as NVRs, CVS, etc. Thus, this research evaluates the performance of streaming protocols in the island network when using IP cameras on encoder standards to lower the load on the network devices.

Therefore, the project aims to evaluate the performance of the island network when using IP cameras on encoder standards. The performance at the network layer have been analyzed for different streaming protocols in this project.

2. History Overview of Video Streaming

When the first commercial streaming services appeared in late 1992, the networks had comparatively huge Round-Trip times (RTT) [7] and provided low data rates than what users can use today. These limitations significantly impacted the design choices relating to transport protocols. At that time, most of streaming video services were intended by using UDP instead of 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 packet delivery, ordering, or error-correction guarantees. In contrast to UDP, Transmission Control Protocol presents reliable connection-oriented services for congestions, flow-control, but doesn't allow the sender to control the send rate. The rate at which packets are delivered depends 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 the same situations without more retransmissions, different error control and concealment techniques [7], forward error correction, and other schemes to mask imperfections were used. Such practices were typically used with early streaming protocols such as (MMS), RTMP [8], and the RTP and its related suite Real-Time Control Protocol (RTCP) and RTSP [3] to mention a few.

2.1 Video Streaming Over the Web

The use of proprietary or UDP-based protocols had major limitations for streaming. The most important among these was that traffic over outwardly initiated UDP or non-standardized protocols is usually blocked by Network Address Translators and firewalls because of security issues with connectionless services using UDP and the risks posed by anonymous protocols. On the other hand, many protocols required the servers to track the client state continually, requiring dedicated infrastructure, and intelligence at the server side. These restrictions and faster internet speeds (that allowed client buffers to be filled quickly) slowly began to outweigh the advantages of using User Datagram Protocol -like protocols. In contrast, the World Wide Web (WWW) 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.

Furthermore, 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 the 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 accessible from many corporations, which every corporation 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 against RTSP, which might enable bypassing several firewalls in IP networks. Microsoft offers smooth streaming, which additionally dispenses with RTSP in terms of hypertext transfer protocol IP technology.

HTTP invented there isn't planning to make it work for video streaming, however, it's been found to be very powerful. Recently, a keep-alive mechanism was introduced, which an association may well be reused for quite one request. Employing a persistent connection can reduce the latency; due to the transmission control protocol, the 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 concerned 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 three completely different protocols hypertext transfer protocol, RTSP and Dynamic Adaptive Streaming over HTTP (DASH) on smartphones. They need calculated switch delay, the delay 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 before it shows the new perspective and restricts 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 victimization network machine on hypertext transfer protocol, mostly video transmission and analysis on the network impairment influence the streaming video. The 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 permit the streaming video to play swimmingly. Authors in [14] have given a QoE instrumentation for video streaming, VLQoE on a smartphone. They need supplemental practicality to the VLC media player to record a group of metrics from the programme, application-level, network-level, and 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 analyzed the user latency of the themes in 2 situations like short and long freezes. They need to find that the user latency for long freezes was exponentially distributed.

In paper [15], the authors have explained QoE's technical and non-technical parameters. 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 connection is the client, and 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 to commands from the client. These protocols are 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 sends a request to the server, which the server starts a communication where the client downloads the video streaming, such as progressive download.

3. Research Methodology

3.1 Methodology of the Project

Firstly, a detailed literature review is studied on the island network, IP camera and streaming protocols such as HTTP, RTSP and then a detailed study on the various schemes proposed by the researchers on the specified protocol. In the next stage, choose different ONVIF IP cameras that are encoded to specific encoder standards for conducting experiments using HTTP and RTSP protocols.

Later, study and analyze the impact of different network emulator tools to inject different delays and packet loss. The performance metrics is necessary to perform the experiments are to be identified by packet analyzer tools. Here, the experiments are conducted with the appropriate stream media server for HTTP and RTSP. Using IP cameras with encoding standards can greatly decrease the bitrate of the IP surveillance video to reduce the bandwidth and storage cost.

During the experimentations, delays and packet loss are introduced by the appropriate traffic shaper in the network via Wireshark. The streaming of the IPCs are conducted over HTTP and RTSP protocols, which the streaming process is done from IP cameras to the clients in a controlled environment. When the stream is received at the client-side, the values are noted at the application and network levels for different protocols.

At the end, the performance of the network on HTTP and RTSP protocols are observed and analyzed. Based on the final results and analysis, conclusions and recommendations are provided between the protocols. Fig. 1 shows a flow chart of all the processes that we followed to do this project.

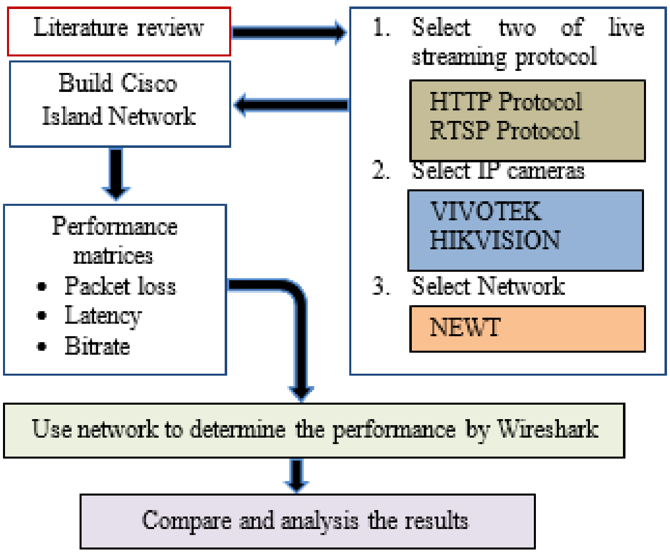


Figure 1: Displays the general flowchart of the project phases

3.2 Island Network of Cisco Video Surveillance

To build a Cisco IP video surveillance system depends on the infrastructure of an IP network to connect all the cameras, NVRs, servers, switches and routers, etc. The network's design has been tested to allow applications to converge on intelligent and resilient infrastructure. Cisco is moving various proprietary systems to a common IP backbone. For the test bed project, the Cisco Island network was built by configuring the C2612-router, then connected to the WS-C2960-24TC-L Switch, cameras connected via LAN cable to the switch, as shown in Fig. 2.

3.3 IP Cameras

In this project, two different camera brands with different encoding standards are used.

- **VIVOTEK PZ81121**

It's a network camera that offers 10x (PTZ), indoor surveillance camera. It supports codec compression for H.264 and MPEG-4 in real-time and streaming over UDP, TCP, HTTP or RTSP, Max. 720 × 480 @ 30 Frame Per second.

- **Hikvision DS-2CD2025FWD-I**

It's 2MP fixed network camera designed for outdoor surveillance applications such as parking lots, monitoring traffic, etc. It supports codec compression for H.265 and H.264 in real-time and streaming over UDP, TCP, HTTP or RTSP, Max. 1920 × 1080 @30 frame per second.

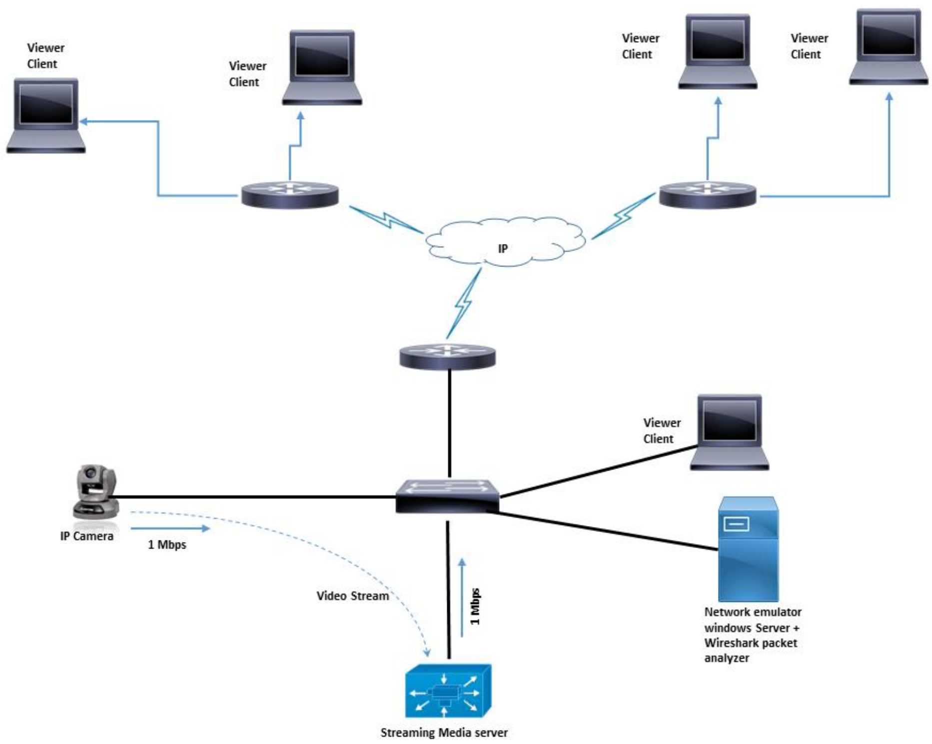


Figure 2: Project diagram

3.4 Experimental Procedure

After the image is captured by the IP camera, processed and compressed have been done inside the camera, the video data is travelled through the network before it reaches the client-side for monitoring. The experimental setup consists of IP cameras, a video streaming server, i.e., HikCentral Stream media server, VLC Media Player at the client, and network emulator. The streaming server is used to send the encoded stream IP camera sequences to the clients. Then, the packets were sent to the client using HTTP, RTSP protocols. NEWT is used to inject loss and variable delays in the network going from server to client. Wireshark is used to capture the traffic at the network layer, and VLC stream statistics are charged at the application layer when the IP camera is streamed via server to the client to verify the packet loss and delay variations at the network layer. The streamer and the VLC Media player are installed on the Windows 7 Professional 64-bit platforms for both server and client. The Wireshark is also installed on Windows 7 Professional.

These are connected using Ethernet cables Cat 5e. The static IP addresses are assigned to Cisco router, switch, server, also the other devices clients, IP cameras. The static IP address for the Cisco router is 192.168.85.1 and for the switch is 192.168.85.2 and the server is 192.168.85.5 with a net mask of 255.255.255.0. The port numbers used for streaming different protocols are HTTP: 8080, RTSP and 8554. Initially, Wireshark was used to check the traffic flowing from IP cameras via server to client when a live video is being streamed, and when the live video was streamed from camera to server then client which the video qualities retrieved on the client were checked by some users.

An experiment is carried out for HTTP and RTSP. The values obtained at the application layer and the network layer are collected. The metrics will collect separately for application layer and network layer. The video is streamed from the IP camera via server to the client at the application layer.

4. Result and Analysis

The project results in evaluating the performance of the island network based on different streaming protocols.

4.1 1% Packet Loss Rate

The study on packet loss was carried out on streaming protocol HTTP and RTSP in the following scenarios individually.

Using NEWT (Network Emulator for Windows Toolkit) could make the study of packet loss on the island network by injecting 1% packet loss to get the result for RTP /RTSP over UDP, HTTP over UDP and HTTP over TCP.

4.1.1 RTP/RTSP over UDP

The client sends RTSP (real time streaming protocol) request to the camera to negotiate about the ports then the video starts streaming to the client, which the video transmits via RTP.

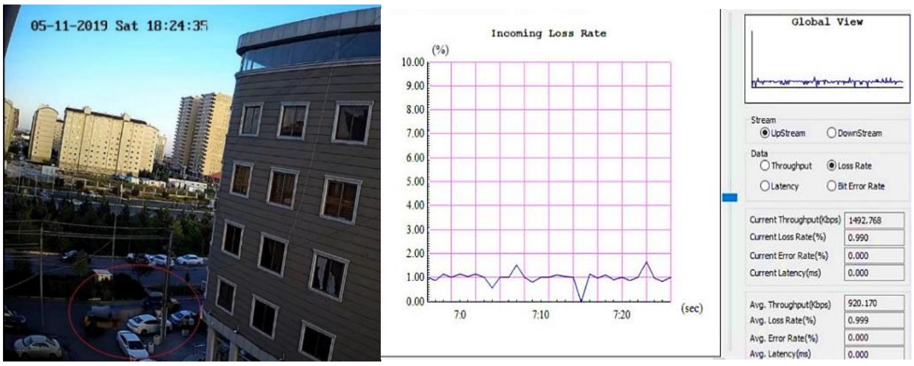


Figure 3: The Loss Ratio (%) graph that injects for incoming by NEWT

Fig. 3 illustrates the packet loss ratio (%) injected by NEWT in the island network between the camera and streaming server. According to the results, it's clear that even though the loss rate is 1%, it shows a high effect on the pictures frame flow when using the RTSP to stream the video on UDP but still streaming video from the camera.

4.1.2 HTTP over UDP

HTTP is a stateless protocol, so when HTTP data request by client, the remote server have been respond to that request via sending the ordered data, which each HTTP request is handled totally independently.

From Fig. 4 (a) and (b), it can be seen that the streaming is stopped; only one captured picture can view on the HTTP around every 20 seconds. Finally, Fig. 5 shows the graph from Wireshark result when the packet loss injected is 1%.

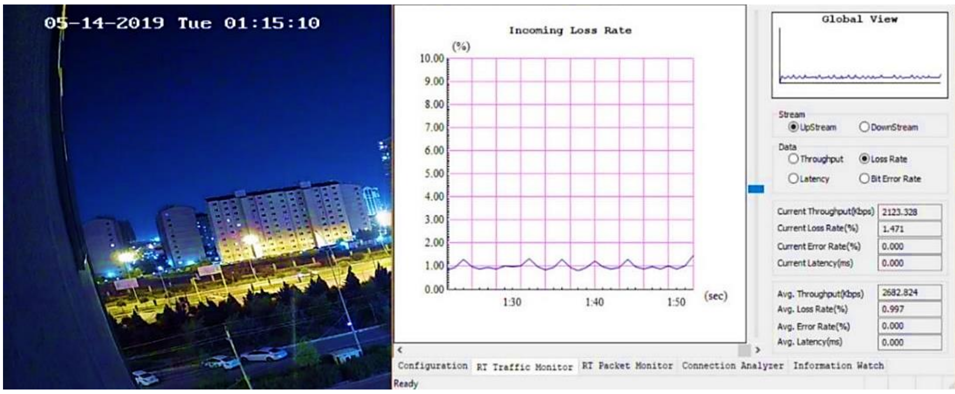


Figure 4 (a): Outdoor street using HTTP with NEWT tool at 01:15:10

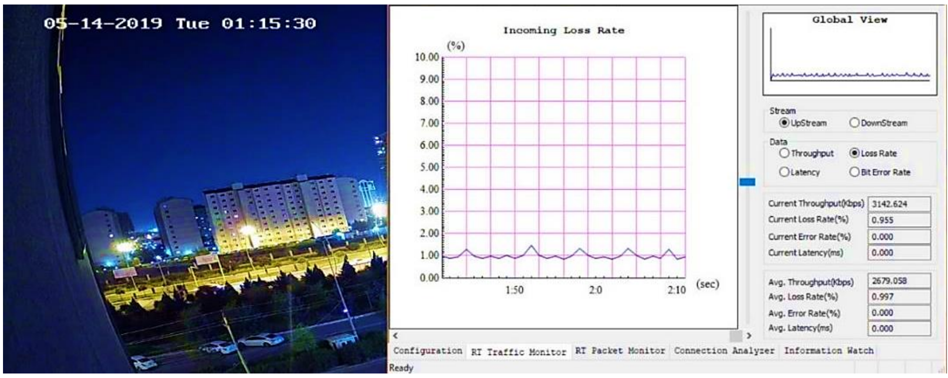


Figure 4 (b): Outdoor street using HTTP with NEWT tool at 01:15:30

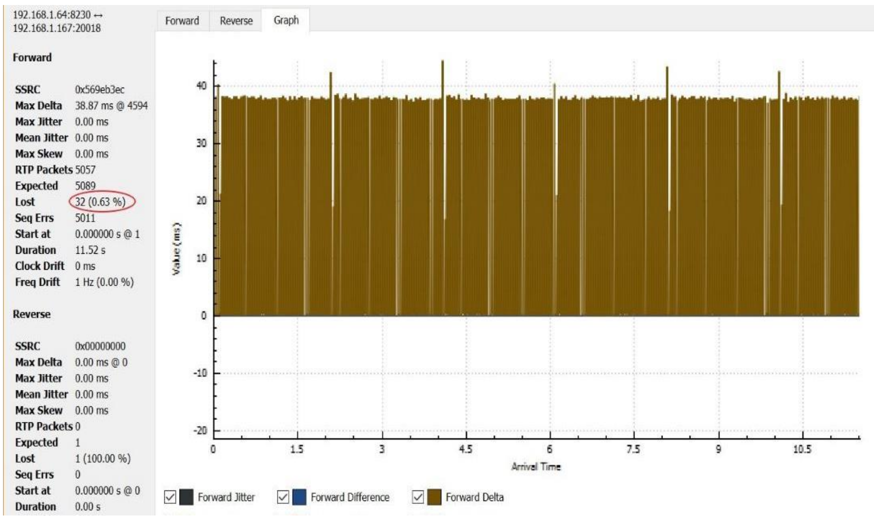


Figure 5: Wireshark graph for the RTP loss

Fig. 5 shows that RTP for 11.52 seconds was 5057 while the expected RTP should be 5089, so the proposed system loss 32 packets that representing 0.63.

4.1.3 HTTP over TCP

Sometimes the TCP RTSP 554 port has been blocked, so it's must use port 80. This port is used for web access and the internet, which video has been transmitted over the port 80 to the clients, which is normal for the internet.

Fig. 6 shows that the graph of loss ratio isn't stable, so the video playing looks normal in the movement even with 1 % loss inject.

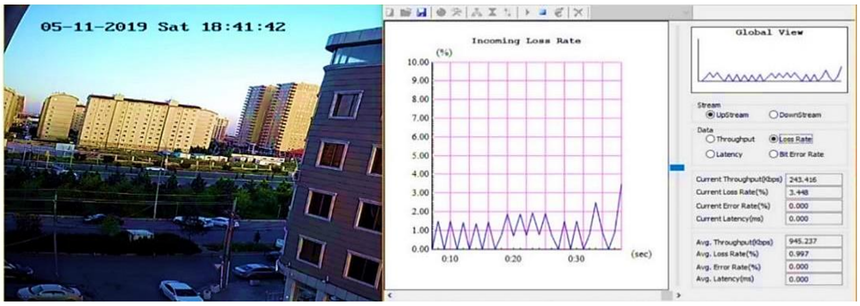


Figure 6: NEWT graph for the loss ratio 1% HTTP over TCP

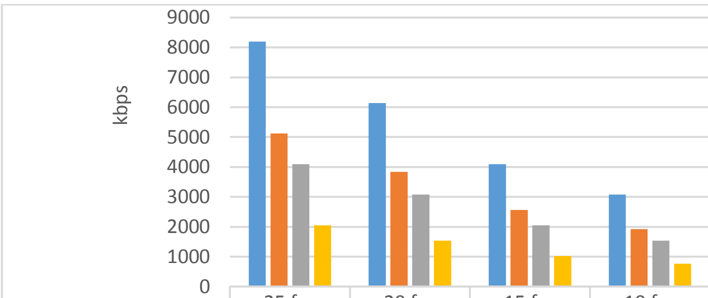
4.2 Bitrate

In this project, the bitrate per second are calculated for different cameras resolutions in the normal scene for two codec H.265/ H.264.

4.2.1 H.265/High-Efficiency Video Coding (HEVC)

Fig. 7 shows the bitrate per second for four different types of resolution. In different frame per second (25,20,15,10) fps, (i) 8MP (3840×2160), (ii) 6MP (3072×2048), (iii) 4MP (2560×1440) and (iv) 1080P (1920×1080).

In Fig. 7, it notices that 8 MP IP camera can stream bitrate 50% less if system reduce the FPS from 25 to 15 fps, and 25% less, if system reduce the FPS from 25 to 20 fps, 62% less if system reduce the FPS from 25 to 10 fps, also same for other resolutions, could reduce the bandwidth and storge requirements.



	25 fps	20 fps	15 fps	10 fps
8MP 3840×2160	8192	6144	4096	3072
6MP(3072×2048)	5120	3840	2560	1920
4MP(2560×1440)	4096	3072	2048	1536
1080P(1920×1080)	2048	1536	1024	768

Figure 7: Comparison of the bitrate (kbps) for codec H.265, different high resolutions and different frame rates

4.2.2 H.264/Advanced Video Coding (AVC)

Fig. 8 shows the bitrate per second of four different types of resolution, in different frames per second (25,20,15,10) fps; (i) 8MP (3840×2160), (ii) 6MP (3072×2048), (iii) 4MP (2560×1440) and (iv) 1080P (1920×1080).

In Fig. 8, it notices that 8 MP IP camera can stream bitrate 50% less if reduce the FPS from 25 to 15 fps, and 25% less if reduce the FPS from 25 to 20 fps, 62% less if reduce the FPS from 25 to 10 fps, also same for other resolutions. That can reduce the bandwidth and storge requirements. Many things may effects on the bitrate like the complexity of the scene if it's indoor or outdoor, also there is a different between the daytime and during night.

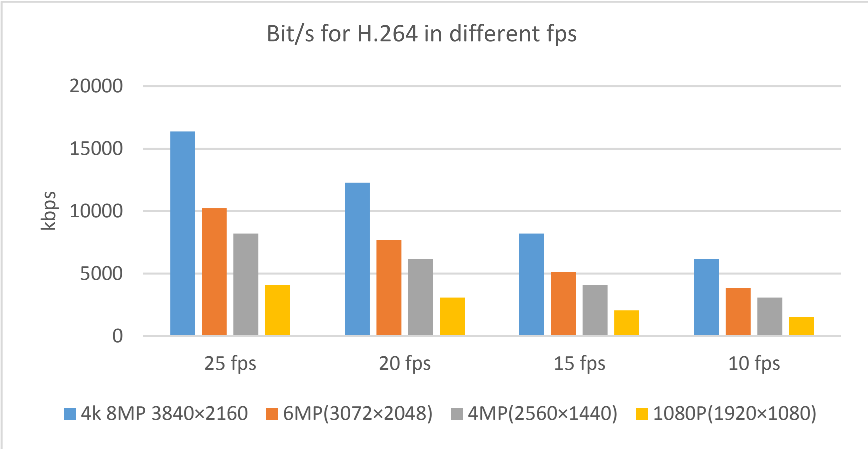


Figure 8: Comparison of the bitrate (kbps) for codec H.264, high different resolutions and different frame rate

4.3 Delay

To calculate the total end to end latency in video surveillance system it should take in considering the time required to encode, transmit, buffer and decode the video to the receiver device, also round-trip time (RTT) of transmission networks. There is three types of delay effective in video system;

(i) Delay by camera
(ii) Delay by network
(iii) Delay by client

For the delay in the camera, it depends on the image capture delay, delay during image enhancement (Image rotation, Resolution, Multiple streams) and compression delay (Complexity of algorithms, Effectiveness of the method, the choice of bitrate VBR or CBR, Buffer latency).

There are three concepts in network surveillance video: Bitrate, Bandwidth, and Throughput. Fig. 9 shows the Bandwidth is the cylinder between the camera and client, it measures how thick the cylinder is. The Throughput measures how much actually data comes through the cylinder per second. The bitrate measures how much data is being carried out to the cylinder per second.

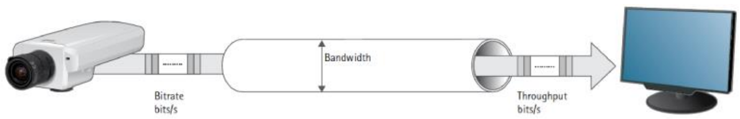


Figure 9: Video network delay factors

The delay in the network is directly proportional to the bitrate and inversely proportional to the bandwidth. The delay in the link depends on three things.

(i) The infrastructure of the network between cameras and the clients which locates the bandwidth,
(ii) The size of data generated for the camera that locates the bit rate and
(iii) Selection of the transport protocol.

The bandwidth depends on the network infrastructure routers, switches, cables etc. If the network bandwidth is increased, the data throughput via the network will be more, and the latency will be less.

Throughput is the actual speed achieved to transfer your data. This depends on whether share the link with other devices. This also depends on the electromagnetic interference on the cables in the connector and the quality of service configured on ports that may limit throughput. Only in live PTZ cameras the delay is more critical while doing the PTZ control. The change in the image should correspond with Joystick movement so interaction can be acceptable. In this project, the system finds that network delay should be less than 50 milliseconds, whereas the total delay with the codec process should be not more than 500 milliseconds.

In fixed cameras, the constant delay is not effective. The sessions can be established, and video can be streaming over the network even if there are a few thousand milliseconds of network delay. Fig. 10 shows the comparison of the average packet number between the HTTP and RTSP in 60 Seconds in different delay injections for the fixed camera. In this research, in-network layer HTTP used TCP while RTSP used UDP as transport protocols. In the data transfer streaming process, packet data is encrypted in the camera before transmission, it is decrypted later in the clients. Thus, the number of packets affected the network bandwidth and storage requirement.

Fig. 10 illustrates the results comparison of average packet number between the HTTP, RTSP. The figure shows that HTTP over TCP packet number is only 65% from RTSP over UDP packet number, that's means HTTP used less bandwidth from the RTSP by 35%. Therefore, HTTP over TCP has less packets when compared to RTSP that means HTTP uses less island network bandwidth. Thereby it can conclude that Hypertext Transfer Protocol is a little superior and authoritative protocol to stream a video when compared to the RTSP protocol.

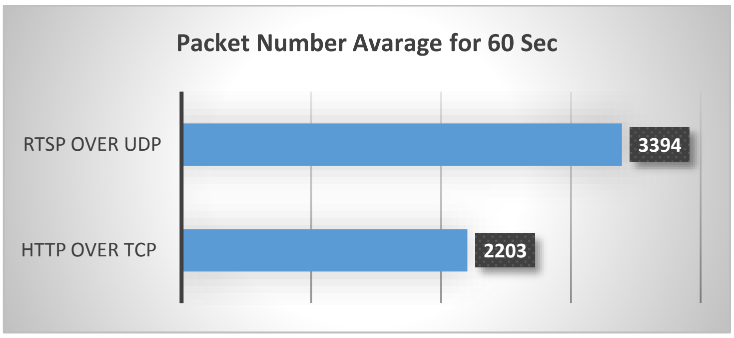


Figure 10: Comparison of average packet number between the HTTP and RTSP

5. Conclusion

IP video surveillance system has become a cornerstone of security in the world; there is no place in the world that haven't of surveillance cameras. The study of streaming protocols for the transmission of these cameras has become an urgent necessity and compares performance between streaming protocols within a live network stated in objectives.

This project has satisfied its objectives by evaluating two different streaming protocols on an island network. The considered protocols Hypertext Transfer Protocol (HTTP) and Real-Time Streaming Protocol (RTSP) for different cases. The evaluation of different streaming protocols for different cases utilizing a set of performance metrics, packet loss, bitrate and delay/latency. The results showed the comparison between the protocols when it injected 1% packet loss to the island network. In average loss rate is 1% it shows high effect on the pictures frame flow when it using the RTSP to stream the video on UDP. However, it is still streaming video from the camera, while on HTTP protocol over UDP the streaming is totally stopped; only one capture picture can view around each 20 seconds. In contrast, HTTP protocol over TCP the video streaming looks normal in the movement even with 1 % loss inject. Comparison of the bitrate on codec H.265, H264 showed that stream bitrate is 50% less when reduce the FPS from 25 to 15 FPS , and 25% less when reducing the FPS from 25 to 20 FPS, 62% less when reducing the FPS from 25 to 10 FPS. So to control the bandwidth on the island network and reduce storage requirements, it's have to manage the frame rate per second depends on the scene that monitoring. Comparison of bitrate (kbps) shows that bitrate at night is only 25% of the bitrate at daytime at indoor environments with normal moving objects. In contrast, a bitrate (kbps) comparison shows that bitrate at night is almost 90% of the bitrate at daytime when the codec H.265, it's about 60% from the daytime when the codec H.264 at outdoor environment with normal moving objects. Comparison of the delay in fixed cameras the constant delay is not effective only in PTZ cameras the delay is more critical during doing the PTZ control the island network delay should be less than 50 milliseconds, whereas the total delay with codec process should be not more than 500 milliseconds. In general, HTTP over TCP has less packets when compared to RTSP that means HTTP uses less island network bandwidth. To sum up, Hypertext Transfer Protocol is a little superior and authoritative protocol to stream a video when compared to the RTSP protocol.

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).