University of Alberta
Edmonton, Canada

A comparative measurement study of the performance of Video Streaming using Windows Media Player

Submitted To: Dr. Mike MacGregor

Submitted By: Adnan Rafique

Dated: August 10, 2006

Abstract

Video streaming plays a vital role in our daily life. As we are sitting on our computer and watching the news, movies, sports or any live or recorded broadcasts, we depend on video streaming. In other words, video steaming is the backbone of today's multimedia communication. Most popular video sites include CNN.com for news stories, Disney.com, sonypictures.com and movies.yahoo.com for trailers of upcoming movies and music.yahoo.com and mtv.com for music videos. All of these sites deliver video content to their user. For that purpose these sites require the streaming servers that can deliver the video streams to their users when it is needed. On a user end a streaming player is needed that can receive, decode and then display the incoming video streams.

Windows Media Player, Real Player and QuickTime Player are the widely used Internet streaming systems.

While talking about video streaming, we focus on compression and communication of video. More precisely, we can say that video streaming is a combination of compression and communication. Communication or delivery of the video contents over the internet is the challenging part because the internet doesn't guarantee the complete and accurate transfer of the data. So, once we have an encoded video, we need to transfer the video stream in such a manner that could maintain the video quality. The main performance issues of concerns in video streaming are delay, jitter, data losses and bandwidth utilization.

In this project, Windows Media Services 9 series in Windows server 2003 is used to stream the video contents from server to the client upon user request. On client side Windows Media Player is used to playback the video streams. We studied the video performance while using different protocol combinations in terms of delay, jitter, packet loss, bandwidth utilization and router CPU-utilization. Video performance was analyzed while generating the video traffic solely and as well as with competing data traffic. Then we have implemented QoS (priority queuing mechanism) on video streaming to study the priority queuing effect on video quality while generating the video streaming along with other competing data traffic.

It was found that video quality is highly degraded when video contents are delivered along with competing data traffic and video keeps degrading its quality as the competing traffic is increased and vise versa. To overcome this issue the priority queuing mechanism was implemented and it was seen that priority queuing maintained the video quality in the presence of competing data traffic.

Protocols selection also plays an important role in efficient video streaming. HTTP and Real Time Streaming Protocol (RTSP) are the major protocols used in video

streaming players. RTSP can be used in combinations with both the TCP or UDP protocols. Here in this project, three protocols HTTP, RTSP-TCP and RTSP-UDP combinations have been used to analyze the video performance.

Table of contents

Acknowledgement

I take this opportunity to express thanks to my supervisor Dr. Mike MacGregor for his continuous guidance and support through out the project. With his enthusiasm, his inspiration, and his great efforts to explain things clearly and simply, he helped me to complete this project successfully. Throughout my project, he provided encouragement, sound advice, good teaching and lots of good ideas.

I would like to thank my family for their unconditional support throughout my degree. I wish to thank my wife for providing a loving environment for me. My brother, my sisters, my brother-in-laws were particularly supportive.I would also like to thank to my friends for their support during this project.

Lastly, and most importantly, I wish to thank my parents for their everlasting love, prayer and support to encourage me to fulfill my dreams to learn and be creative. To them I dedicate this project report.

Introduction

Due to increase in multimedia communication, video streaming is increasing its demand in this era. Video streaming is the process of delivering video contents to streaming player for immediate playback. Video streaming is better than downloading because the server delivers the video content to the clients when it is needed and the delivered video contents are not stored inside the client device. Windows Media Services 9 Series consists of fast streaming which combines the advantages of both streaming and downloading. Through video streaming, users can watch the video content at any point in the stream and fast forward, pause or go back in the stream.

Video streaming faces many challenges. The network between the streaming server and destination has a major impact on video quality. Performance issues like delay, jitter, packet losses and throughput utilization have a prominent effect on video streaming. And any interruption and congestion in the network can severely degrade the video image

Performance is the major concern in video streaming. Therefore, there is a need to develop certain standards that can assure the high quality video communication. For that reason, certain QoS measures have been developed and applied in packet communication system to make the video communication highly reliable and accurate. There has been a lot of research going on in this aspect. J. Apostolopoulos, W Tan and S. Wee [1] studied and described the major challenges in video streaming over internet.

They describe, "Video streaming over internet is very challenging because internet does not guarantee QoS. It has no control over bandwidth, delay and losses. Therefore, streaming system should be designed in such a way that it must have ability to provide the high quality video over the internet while dealing with time varying factors such as: bandwidth, delay and losses."

Yubing Wang, Mark Claypool and Zheng Zuo [2] empirically evaluated the RealVideo performance across the internet and found that "the average RealVideo clip streamed over the Internet has good quality, playing out at 10 fps. Users connecting to the Internet with modems and/or slow computers still have their PC or their network connection as the video performance bottleneck, while newer computers connecting to the Internet via DSL or Cable modem achieve even slightly better performance than corporate network connections to the Internet. This suggests that increasing broadband connections for home users are pushing the bottlenecks for video performance closer to the server."

V. Mariappan and P. Narayanasamy [3] studied the video streaming over WLAN by Implementing of SCTP over IEEE 802.11a and showed that the use of SCTP for video transmission overcomes the delay experienced by the end hosts, flow control and congestion control problems.

A. Mena and J. Heidemann [4] briefly studied and examined the traffic generating from a single Internet audio service using RealAudio. They found that UDP is the leading download transport protocol, suggesting non-TCP congestion control. Chesire, A.

Wolman, G. Voelker, and H. Levy [5] studied RTSP packets traffic. There are three major Internet streaming systems in use today: Windows Media Technology (WMT) from Microsoft [6], RealNetwork streaming technology from Re-alNetworks [7], and QuickTime from Apple [8]. To the best of my knowledge, there has not been a video performance study using the Windows Media Players.

In this project, we studied video streaming performance using Windows Media Player while using different transport protocols combinations. The performance study was carried out in three phases.

First we studied the video performance without any competing traffic. In this case, performance and suitability of Windows Media Player was studied in terms of following issues:

1- Delay
2- Jitter
3- Throughput
4- Frame rate
5- Packet loss
6- CPU utilization

After analyzing the video performance solely, competing data traffic was introduced and above mentioned parameters was studied. In this scenario, we studied the effect of competing traffic on video traffic and analyzed the video behavior in terms of video quality.

Based on the results drawn, we implemented some QoS to study the effect of video performance in the presence of competing data traffic. By implementing QoS mechanism we were able to make the video streaming more reliable and achieved the required video quality.

Before, we proceed any further; we will have a glance at organization of the report.
The report is organized in 7 chapters.

Chapter-1 It presents the introduction on Windows Media Technology: mechanism and principles.

Chapter-2 It presents Procedure of Video Streaming Performance Study being done in this project.

Chapter-3 It presents the Phase One of a project: Video performance analysis without any competing data traffic using Windows Media Player.

Chapter-4 It presents the Phase Two of a project: Video streaming performance analysis in the presence of competing data traffic

Chapter-5 It presents the Phase Three of the project: Implementing Priority Queuing (PQ) mechanism while transferring the video contents along with other competing data traffic.

Chapter-6- It presents the conclusions and future directions

Chapter-7 It provides the references obtained to write this report.

Chapter-1

Windows Media Technology: mechanism and principles

"Windows Media Services 9 Series was updated for the Microsoft Windows Server 2003 operating systems with Service Pack 1 (SP1), and for the new x64-based versions of Windows Server 2003, adding advanced streaming functionality and native 64-bit support for even higher scalability. Now more than ever, it's the industry's most powerful streaming media server. [9]"

Windows Media technology shares the following characteristics:

- The principal transport protocol can be UDP, TCP, or HTTP.
- The client uses a buffer to stabilize the variations in packet arrival rate.
- The streaming server tries to transmit the video contents at a constant bit rate, i.e. the encoded bit rate of the streaming media. It usually transmits a cluster of data, waits for a fixed duration and then sends another cluster of data.
- When available bandwidth changes, the player selects among a fixed set of bit rates for the streaming media, and the server serves the selected stream. The streaming media server is a multi-CBR source.
- The selected bit rate is determined by the player which typically uses the data arrival rate to determine when to upgrade or downgrade the stream.

The streaming server sends data at the bit rate selected by the client. The client maintains a buffer into which data is put upon arrival from the network, and from which data is pulled for rendering at the constant bit rate of the selected stream. The client decides when to upgrade or downgrade based on the stability of the data arrival rate into the buffer.

1.1 Fast Streaming

Windows Media Services 9 Series [10] uses fast streaming to deliver the video contents to its users. In fast streaming, a video server delivers the contents as fast as it can so that the client can playback the contents as quickly as possible. The streaming player downloads and buffers a small portion of the contents as fast as the network allows before the contents starts to play. Once the buffer has been established, the streaming server slows down the stream according to available bandwidth and speed of the streaming player.

Windows Media server works with Windows Media Player to detect network conditions and adjust the properties of the stream automatically to ensure the maximum playback quality of the video contents. For that purpose, video contents are encoded with streams of multiple bit rates.

Windows Media Services 9 Series in Microsoft Windows Server 2003 involves the following steps to deliver the video contents to the users.

- Save the video contents to the Windows Media Server.
- Create the publishing points.
- Provide clients with access to the contents by either creating the announcement file or by providing users with the URL of the publishing point.
- Embed the announcement file or URL in a web page or send it in an e-mail message to the clients.
- Clients click on the link or the announcement file and use the URL to connect to the video stream.

Note: Publishing point is defined as "a virtual directory used for storing content that is available to clients or for accessing a live stream. Clients reach a publishing point through its URL."

1.2 Protocols overview

Internet is comprised of millions of different networks running a wide range of hardware and software combinations [11]. As a result, the ability to stream video contents to the users depends on several well constructed protocols. Windows Media technology uses the following protocols to stream the video contents:

- Real Time Streaming Protocol (RTSP)
- Microsoft Media Server (MMS) protocol
- Hypertext Transfer Protocol (HTTP)

Windows Media Services use the control protocols plug-ins to manage the use of these protocols. The control protocol plug-in receives the incoming user request, decides what action is to be taken in response to that request (for example, to start, pause or stop the stream), translates the request into a command and then passes the command to the server.

Control protocol plug-ins can also notify the clients if there is an error condition or a change of status. While the control protocols plug-ins handle the high level exchange of data, the basic networking protocols such as Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are used to manage more fundamental tasks such as network connectivity and packet error correction. Microsoft Media Server (MMS) protocol and Real Time Streaming Protocol (RTSP) protocols can be used in combination with both User Datagram Protocol (UDP) and Transmission Control Protocol (TCP) protocols.

1.2.1 Protocol rollover

"The ability of Windows Media Services to choose the right protocol for a client depending on its environment is known as protocol rollover. [12]"

Protocol rollover is useful if the clients are connecting to the server through different types of network or there is a firewall between a client and the server. Windows Media server uses protocol rollover to create the best possible connection with the client. While connecting to the server, the client first sends information regarding its type and protocol to the server. The Windows Media server then compares that information to the protocols that are enabled and uses the best protocol to establish the connection.

1.2.2 Hypertext Transfer Protocol (HTTP)

HTTP [13] can be used to stream the video contents from Windows Media server to the clients running the Windows Media Players. HTTP is useful for the clients that receive video contents through the firewall because HTTP is set up to use the port 80, which is not blocked by most firewalls. If HTTP is used to receive streams from the server, protocol rollover is not used.

1.2.3 Real Time Streaming Protocol (RTSP)

RTSP [14] can be used to deliver video streams to the computers running Windows Media Players. It provides ability to the clients to stop, pause, rewind and fast forward the contents during playback. RTSP is application-level control protocol that works in tandem with Real Time Protocol to deliver contents to the clients. When the client connects through RTSP using URL (for example, rtsp: // server_name/publishing_point_name/file_name), RTSP directs the RTP to deliver the video streams using the UDP or using the TCP on network that does not support the UDP. RTSPU (UDP based protocol) is the preferred protocol for streaming.

We can also force the server to use a specific protocol; we can mention the protocol to be used in the announcement file.

The user can also mention the protocol in the URL address as:

- rtspu: //server_name/publishing_point_name/file_name
- rtspt: // server_name/publishing_point_name/file_name

1.2.4 Microsoft Media Server (MMS) protocol

Microsoft Media Server protocol [15] is the proprietary streaming media protocol developed by Microsoft for earlier versions of Windows Media Services. MMS can also be used to stream video contents. It also provides ability to the clients to stop, pause, rewind and fast forward the contents. When Client connects to the stream through MMS using the URL (mms: // server_name/publishing_point_name/file_name), player uses the protocol rollover to select the best protocol to get the video streams.

MMS can be used as MMSU and MMST. MMSU is the UDP based protocol and MMST is the TCP based protocol. MMSU is the preferred protocol for streaming. We can also force the server to use a specific protocol; we can mention the protocol to be used in the announcement file.

The user can also mention the protocol in the URL address as:

- mmsu: //server_name/publishing_point_name/file_name
- mmst: // server_name/publishing_point_name/file_name

Note:

Windows Media Services 9 Series uses RTSP instead of MMS protocol. When client tries to connect through MMS the server automatically uses RTSP.

Chapter-2

Procedure of Video Streaming Performance Study

In this project, the video performance is studied with and without the performance of other competing data traffic. Following are the steps involved in methodology of the project.

2.1- Equipment used

1- Windows Media Services 9 Series in Microsoft Windows Server 2003 operating system (AMD Athlon, 1.99 GHz, and 512 RAM) is used to stream the video contents.
2- Windows Media Player is used to play back the video streams on client (Intel Pentium M, 1.8 GHz, and 256 RAM) side.
3- Enterprise LAN meter is used to generate the competing data traffic.
4- DominoFE DA-350 Internetwork Analyzer is used to capture the data packets and calculating the delay and jitter.
5- Cisco Catalyst switches -3500 series.
6- Cisco Routers – 2600 series.
7- Video file length 33 minutes
8- Video data rate of 308 kbps + header size
9- Video size is 86 Mbps

2.2- Network Setup

1- Windows Media Server in Microsoft Windows Server 2003 operating system and Media Player are attached to the router Ethernet port 0 via Cisco Catalyst switch 100 Mbps link.
2- Cisco routers are attached to each other through HDLC via 1Mbps serial link.
3- Enterprise LAN meter is attached to switch to generate the competing data traffic.
4- DominoFE Internetwork Analyzers are attached to both client and server to capture data packets.

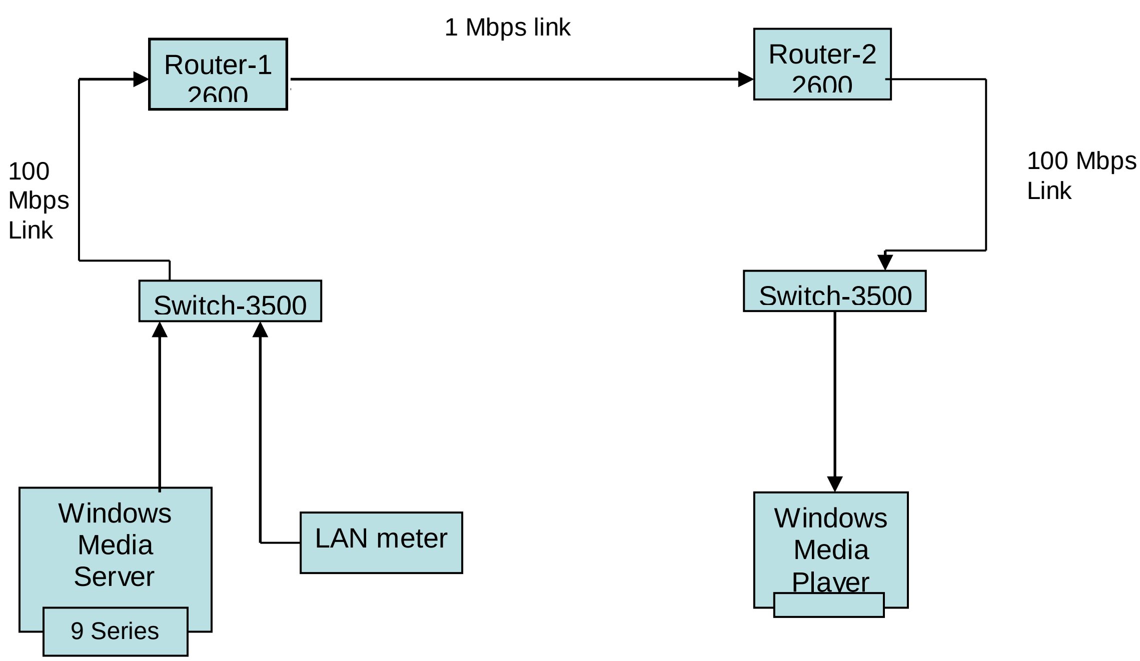
2.3- Procedure

1- First of all, the video contents were streamed across the network and performance parameters i.e. delay, jitter, throughput utilization, frame rate, packet loss and router CPU-utilization were studied. Windows Media Server was used to stream the video file of 33 minutes long and 86 Mbps to the client for immediate play back. And performance parameters were studied.

2- Competing data traffic was introduced and then again performance of video was analyzed as follows:

- Keeping the data traffic below and equal to the link capacity. In this case competing traffic was sent at different speeds i.e. 340 kbps, 508 kbps, 672 kbps, 840 kbps and 1 Mbps.
- Keeping the data traffic above the link capacity. In this case competing traffic was sent at different speeds i.e. 1.25 Mbps, 1.05 Mbps, 1.08 Mbps, 1.1 Mbps, 1.13 Mbps and 1.16 Mbps.

3- Implemented Priority Queuing(PQ) mechanism to the router Ethernet port to analyze the effect of competing data traffic over video traffic keeping the competing data rate above the total link capacity.



Network Setup

Figure 2.1

Chapter-3

Video performance analysis without any competing data traffic using Windows Media player

Windows Media Services 9 Series server is used to stream the video clip to analyze the video quality while using different protocols combinations and clock rates. First we analyzed the video streaming without any competing data, so that we could find that how video traffic behaves and how the performance issues affect the video's quality. It was noticed that if there is no other competing data traffic, the Windows Media Player plays back the video with high quality and accuracy according to the provided link speed. The protocols combinations used in this project are as follows:

1. HTTP
2. Real Time Streaming Protocol (RTSP-UDP)
3. Real Time Streaming Protocol (RTSP-TCP)

Note: HTTP and RTSP-TCP behaves similarly.

Following are the major performance issues that have been studied to analyze their effect over video streaming.

3.1-Delay

Packet delay is an important performance parameter. It has a vital impact on video quality. Packet delay can be defined as "The time taken by a data packet from server to the client." Under controlled conditions, when there is no competing data traffic, packet delay remains constant. It does not degrade the video quality. Following is the behavior for the delay observed in the project.

In bottleneck link, delay value is based upon three factors.

1- Distance between server and client
2- Packet injection rate / clock rate
3- Queuing delay

- **Distance between server and client**

The distance between server and client has no major impact over delay value because in communication networks, we can send the data at the speed of light. Therefore, this factor can be ignored in case where the distance between the source and destination is not much (as in this project).

- **Clock rate**

Clock rate is defined as "the rate at which the router-1 injects the packet in the serial link." Clock rate has a main impact on delay values and affect the video quality. By setting the clock rate equal or greater than the incoming video data rate, we can get the video streams without any packet drop and with good quality and higher frame rate (up to 30 fps).

"By increasing the clock rate, we can decrease the packet delay .It doesn't increase the video quality and frame rate. And by decreasing the clock rate lower than the video data rate (required bandwidth to stream the video clip accurately and with high quality); there is a huge packet drop at router-1 serial port. As a result, router-2 can not receive the video packets and ultimately client is not able to play the video properly."

- **Queuing Delay**

Another important factor in delay model is "queuing delay". Router use queues to handle the packets. The coming packets are placed into queues and then transferred to the network. Video server transmits the video packets in clusters. When these packets are reached to server, these are placed in the queue and then transferred to their destination. Then amount of time taken by a packet to travel through the router is affected by queue length. (i.e. the number of packets present in the queue).

In this project following scenario was used to study the delay mechanism. The server is sending the video traffic with rate of 100 Mbps. The link speed between router-1 and router-2 is 1.544 Mbps. Router-2 transmits the received video packets to the client with the rate of 100 Mbps. And the router-1 injects the packets to the serial link at the rate of 1 Mbps (i.e. the clock rate).

Queue

||||||||||||| 1.544 Mbps

Server-------------------→ Router1-----------------------------→Router2---------------→ Client

100 Mbps Injection rate 1Mbps 100 Mbps

As a result, queue starts building up at Router1 because Router-1 is receiving packets with rate of 100Mbps but forwards these packets with rate of 1Mbps only. Delay is measured by implementing HTTP, RTSP-TCP and RTSP-UDP separately. In case of HTTP, RTSP-TCP data packets, the delay between server and client is 12.14ms. As the server is transmitting the clusters of packet and due to difference in link speeds, each coming pack has to wait for extra 12.14ms in the queue before transmission. As a result, delay is increased by 12.14 for every coming packet.

In case of RTSP-UDP data packets, the delay between server and client is 6.27ms which is half of the delay of TCP packets. As a result each packet has to wait for extra 6.27ms in the queue before transmission.

This behavior can be seen in the Figure below.

The queuing delay is mentioned below:

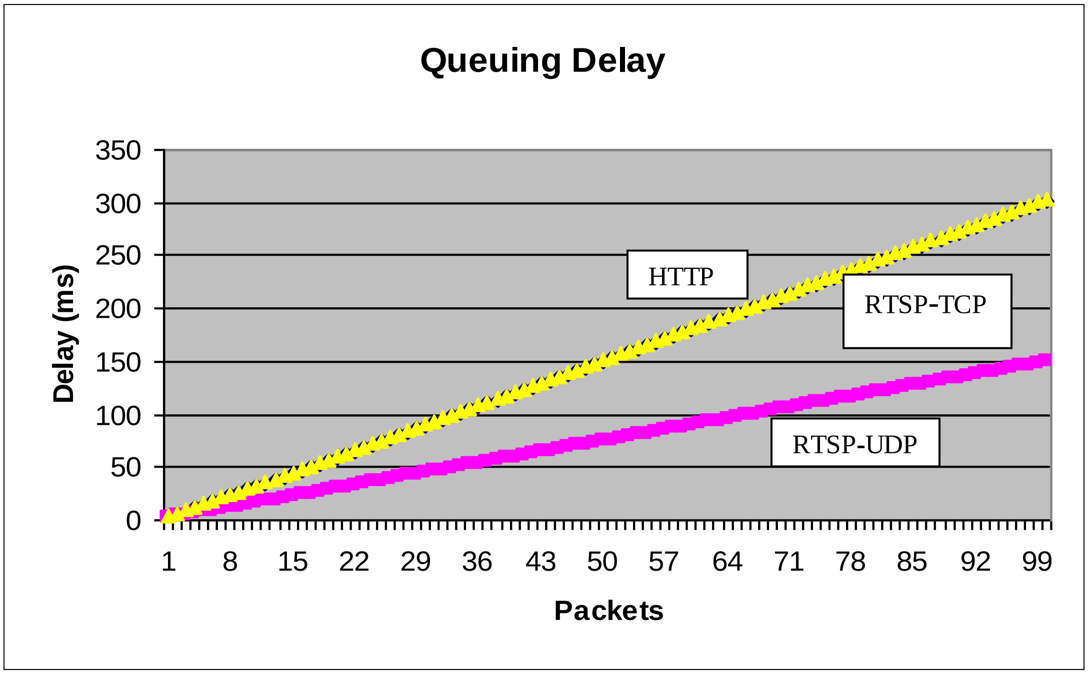


Figure 3.1

Actual Delay

It is an actual time taken by a video packet from video server to the streaming player excluding any queuing delay. A constant delay value was observed while streaming the video from server to the client. The behavior is mentioned below:

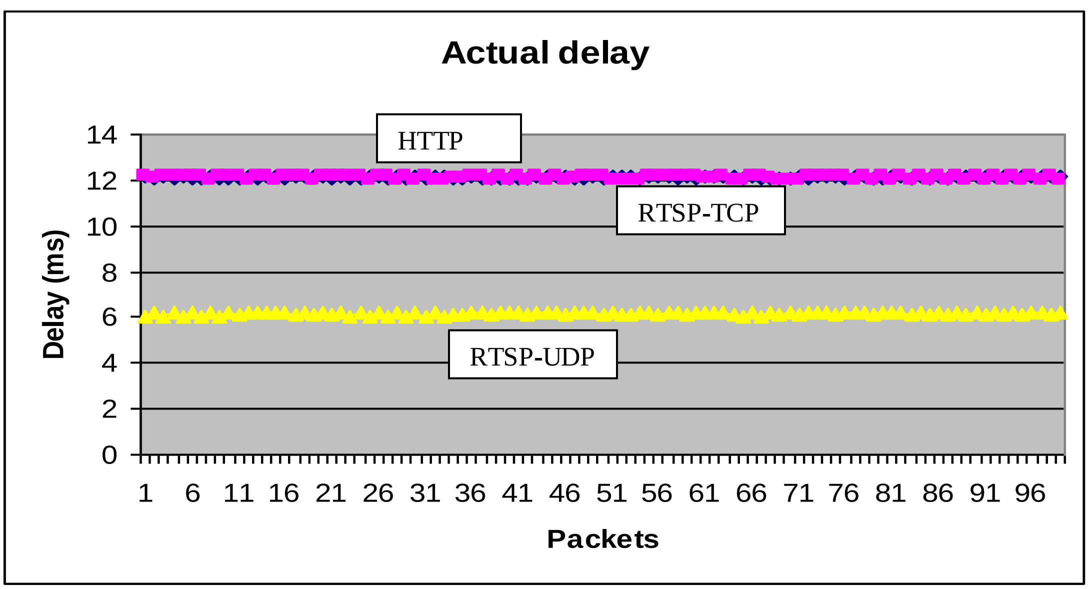


Figure 3.2

This Figure shows that the delay value remains constant and all three protocols (HTTP, RTSP-TCP and RTSP-UDP) behave in same fashion, when video contents are streamed without any competing data traffic.

Delay Calculations:

Delay is calculated by measuring the time when a data packed leaves the video server and reaches to the client. In this project Dominos were used to calculate the delay between video server and the client. Delay was calculated by using HTTP, RTSP-TCP and RTSP-UDP separately.

1- Delay calculation by using HTTP and RTSP-TCP

Data packet size = 1518 bytes or 12144 bps
Packet injection rate = 1Mbps
Delay can be calculated as:
Delay = data packet size / bottleneck link speed
= 12144 bps / 1 Mbps
= 0.012144
Delay = 12.14 ms for each packet

2- In case of RTSP-UDP

Data packet size = 784 bytes or 6272 bps
Packet injection rate = 1 Mbps
Delay can be calculated as:
Delay = data packet size / bottleneck link speed
= 6272 bps / 1 Mbps
= 0.006272
Delay = 6.27 ms for each UDP packet.

3.2- Jitter

"Jitter is defined as the measurement of variation in the arrival time of data packets."

An ideal network is always jitter free. It means, at the receiver the time between each arriving packet is the same. But in reality it does not happen. In case of normal data like e-mail or file downloading, jitter does not have greater impact, but for video and audio communication jitter is very important.

A high variation in jitter can degrade video quality. To accommodate the variation in interarrival time of data packets, a buffer is used to reduce the effect of jitter at the receiver end.

The effect of jitter is observed on video streaming. It was noted that when there is no competing data traffic, the video contents are streamed with very minute jitter. And it does not have any major impact on video streaming. The jitter is calculated by implementing HTTP, RSTP-TCP and RSTP-UDP separately. A very minor variation in jitter is observed in this case. Jitter behavior is shown below

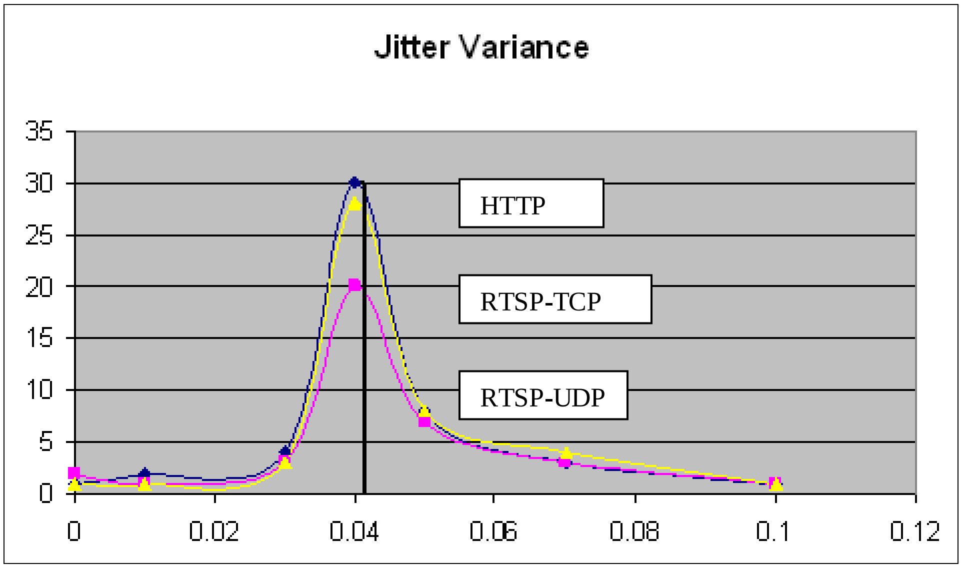


Figure 3.3

Mean jitter is calculated by following formula:

$$\text{Mean} = \mu = \frac{\Sigma X}{N}$$

Variance is calculated as:

$$\text{Variance} = \sigma^2 = \frac{\Sigma (X-\mu)^2}{N}$$

Where,

ΣX….is sum of all the observation values

N……is the number of observations

3.3-Throughput

Throughput is the number of bits per seconds transferred between two devices in the network. Utilization is the ratio between the actual and maximum network throughput.

In this case, when there is only video traffic running on the single link, the maximum utilization is achieved according to the link capacity. Throughput can be calculated as below:

Throughput calculations

1-In case of HTTP and RTSP-TCP

TCP packet size = 1518 bytes or 12144 bits

This 1518 bytes packet includes:
Ethernet Header = 18 bytes
IP header = 20 bytes
TCP header = 20 bytes
Total overhead = 58 bytes
So, total payload in each packet = 1518 -58 = 1460 bytes

2- In case of RTSP-UDP

UDP packet size = 784 bytes or 6272 bits
This 784 bytes packet includes:
Ethernet Header = 18 bytes
IP header = 20 bytes
UDP header = 8 bytes
Total overhead = 46 bytes
So, total payload in each packet = 784 -46 = 738 bytes

1- To calculate the through put we consider first 100 TCP packets
Total packet length for these 100 TCP packets = 151800 bytes or 1214400 bits
Time taken to transmit these 100 packets = 1.2 seconds
Throughput = 1214400 / 1.2
= 1.01 Mbps

2- To calculate the throughput we consider first 100 UDP packets
Total packet length for these 100 UDP packets = 78400 bytes or 627200 bits
Time taken to transmit these 100 UDP packets = 0.62 seconds
Throughput = 627200 / 0.62
= 1.0 Mbps

Maximum throughput is achieved when there is only video traffic is running on the link. As the link capacity is 1 Mbps, so the achieved throughput is the maximum i.e. 1Mbps

3.4- Packet Drop and Frame rate

Packet drop has a huge impact on video frame rate. Video quality is based on frame rate. Frame rate of 15 fps is required to get the good video quality. In our experiment, video of 308 kbps data rate with 30 fps is used. We had a 1.544 Mbps link available and clock rate was 1 Mbps. In this case we can transmit the video data up to rate of 1Mbps without any packet drop. At this link speed, zero packet drops was observed and client had been able to achieve the frame rate of 30 fps.

Following is the graph, showing the achieved frame rate without any competing data traffic:

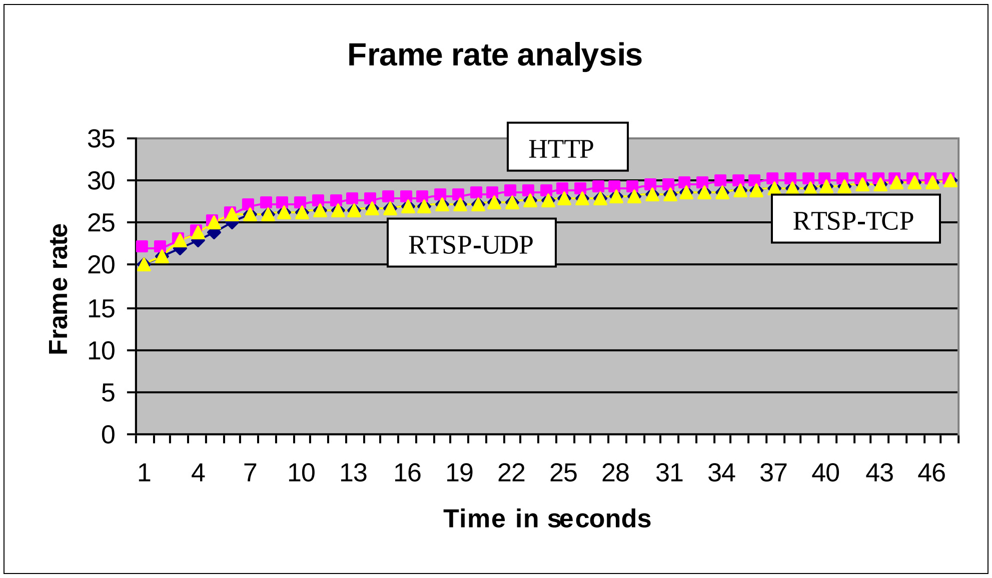


Figure 3.4

It was found that when there is no other competing data traffic; the client achieves the frame rate up to 29-30 fps in less than 1 minute after completing the buffering stage.

3.5-Router CPU- utilization

Router CPU utilization was observed by implementing HTTP, RTSP-TCP and RTSP-UDP separately. In each case CPU utilization was recorded after every 1 minute for 33 minutes i.e. total video clip length. Following figure presents the router CPU utilization.

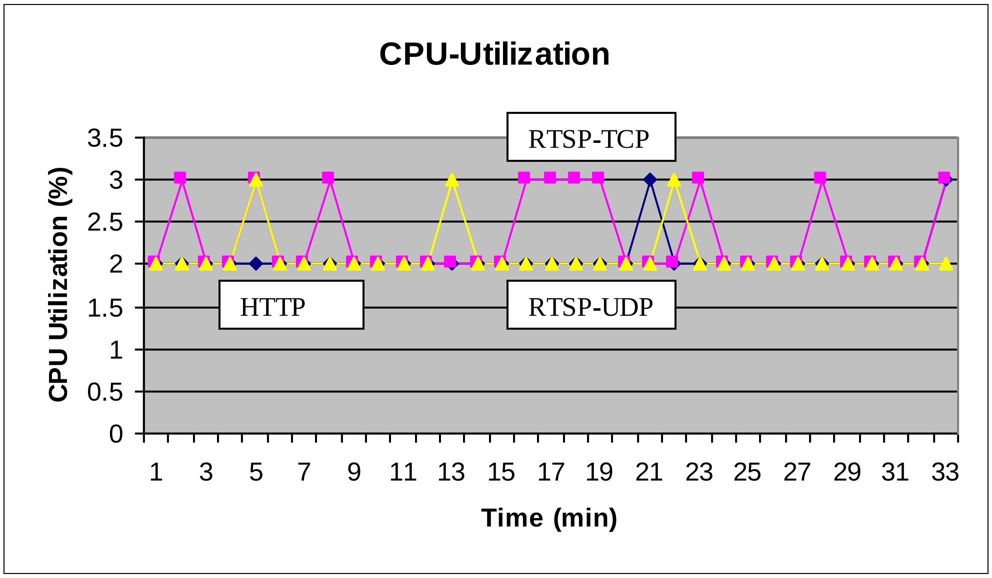


Figure 3.5

This figure show that when there is only video traffic, the router CPU utilization remains low and varies between 2-3 %. It does not load up the router. The three protocols HTTP, RTSP-TCP and RTSP-UDP behave similarly in terms of router utilization.

3.6 Summary

A constant delay is observed while streaming the video without any competing data traffic. Hence, there is no significant increase in the delay except the queuing delay. This increase doesn't deteriorate the video quality and client plays back the video smoothly and accurately without any deterioration in video quality.

Jitter remains constant when there is no competing data traffic. A very low variation in jitter is observed in this case. This small variation in jitter did not degrade the video quality and client received the high quality video contents. Maximum throughput is achieved when there is only video traffic is running on the link. As the link capacity is 1 Mbps, so the achieved throughput is the maximum i.e. 1Mbps. Frame rate of 30 fps is achieved without any packet drop.

Router CPU utilization remains low and varies between 2-3 % when there is no other competing data traffic. In this case, router is not loaded up. The three protocols HTTP, RTSP-TCP and RTSP-UDP behave similarly in terms of router utilization.

Chapter-4

Video streaming performance analysis in the presence of competing data traffic

The competing data traffic has a great impact over video streaming performance. The performance issues behave differently when video contents are streamed in the presence of other competing data traffic.

If the traffic (video + data) remains equal or below the link rate, then all protocols behave in a similar fashion and there is no adverse effect on video quality in terms of packet delay, jitter and packet drops. But when the data traffic is increased than the available link speed, then video quality is highly suffered. Video quality is deteriorated and decrease in frame rate and increase in packet drop, delay and jitter is observed as the data traffic is increased and vise versa. The behavior of streaming protocols is also changed in the presence of competing data traffic.

In this case, 1 Mbps link was used. Video data rate was 380 bps. In this case, we can send another 620 bps of data over this link without affecting the video traffic performance. But when the incoming traffic is increased than the link capacity (1Mbps), it starts dropping the video data packets along with other competing data traffic. As a result, the video quality is degraded and clients receive the video contents with poor quality. The video quality keeps degrading as the competing data traffic is increased and vise versa. Video performance issues were analyzed in the presence of competing data traffic to study the behavior of video streaming.

4.1-Delay

Competing data traffic has a major impact over delay. As the competing traffic increases, delay is increased. Delay is measured and analyzed in two ways.

1. Keeping the total traffic (video + competing data traffic) below and equal to the threshold level (total link speed).

In this scenario, the competing data traffic was increased at different rates. It was found that when the total traffic was increased, the total delay value is increased according to the data rate and vise versa. It was observed that when the total traffic was kept below or equals the link speed, the video quality was not degraded. This increase in delay did not affect the video quality adversely. This is shown in the figure 4.1 and 4.2

2. Increasing the total traffic video + competing data traffic) above the threshold level (total link speed).

When the total traffic was kept above the link speed, the delay was increased and this increase had a severe effect over video quality. As we kept increasing the data traffic above the total link capacity, the video streaming performance was decreased.

This behavior can be seen in figure 4.1.and 4.2.

1- By implementing TCP (HTTP and RTSP-TCP)

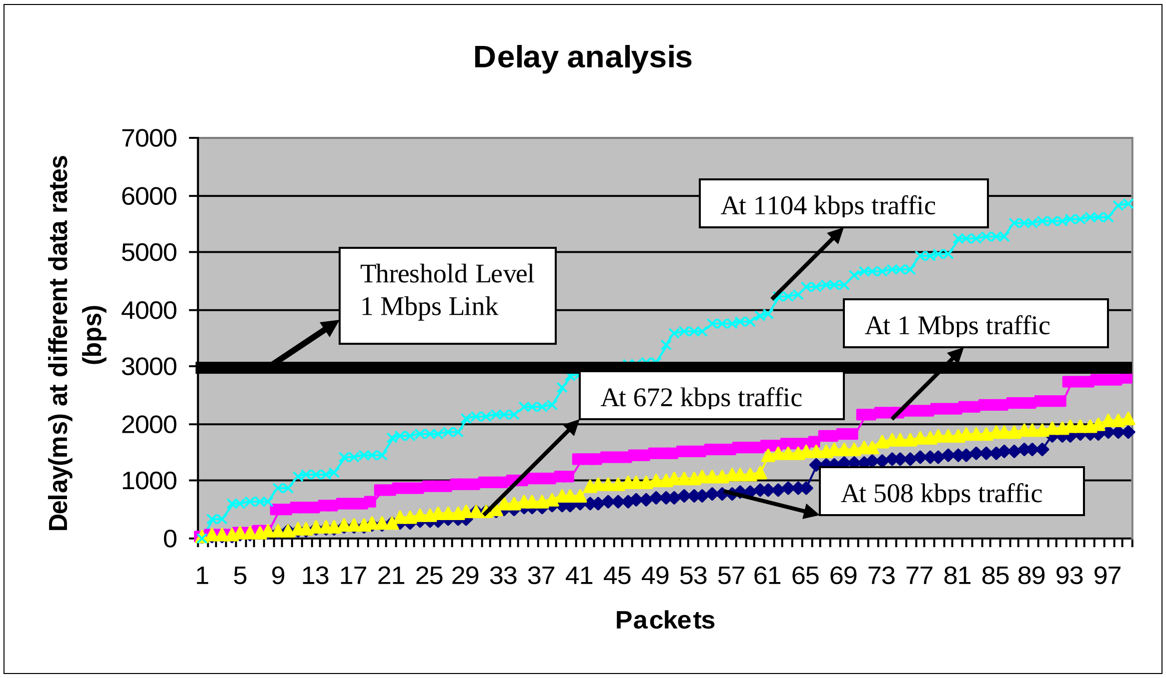


Figure 4.1

2- By implementing RTSP-UDP

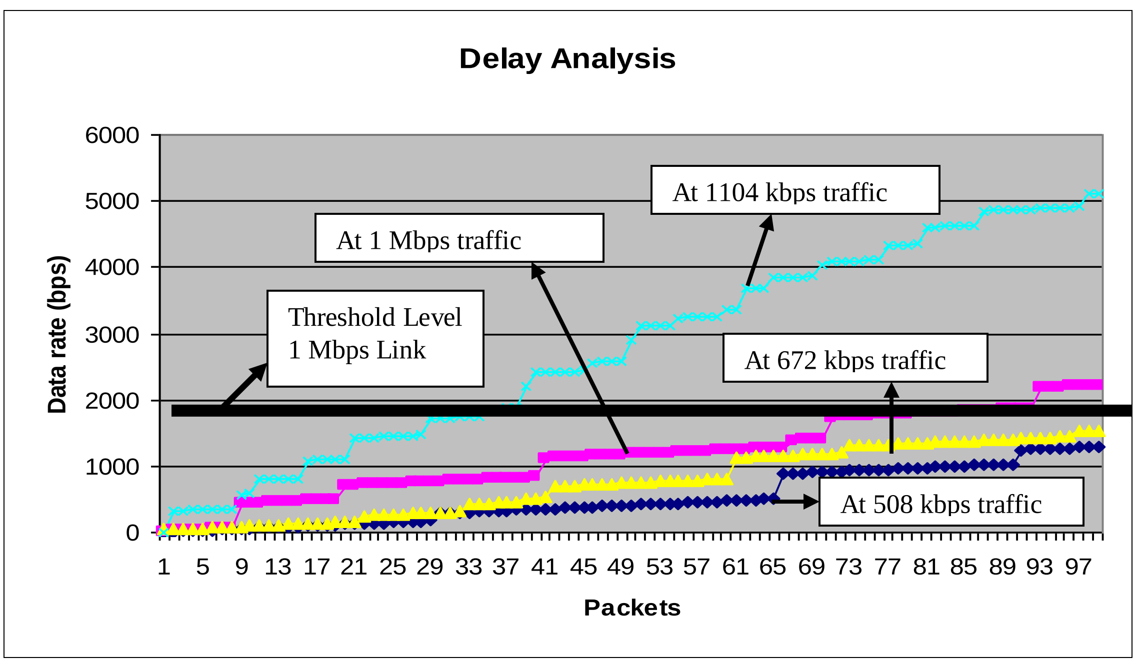


Figure 4.2

4.2-Jitter

Competing data traffic affects the Jitter. As the data rate is increased the Jitter variance is increased. When we kept the total traffic rate below or equal the link speed, variation in jitter was not observed. But as the total traffic rate exceeded the threshold level, a significant change in jitter was observed. A high variation in jitter was noticed, due to which client could not be able to play back the video properly.
Note: All three protocols, HTTP, RTSP-TCP and RTSP-UDP behave in the same way. Jitter behavior at different data rates is given below.

1- At data rate of 672,000 bps:

The figure below shows that at the data rate of 672000 bps, some variation in jitter was observed but this variation was not so severe that it could degrade the video quality.

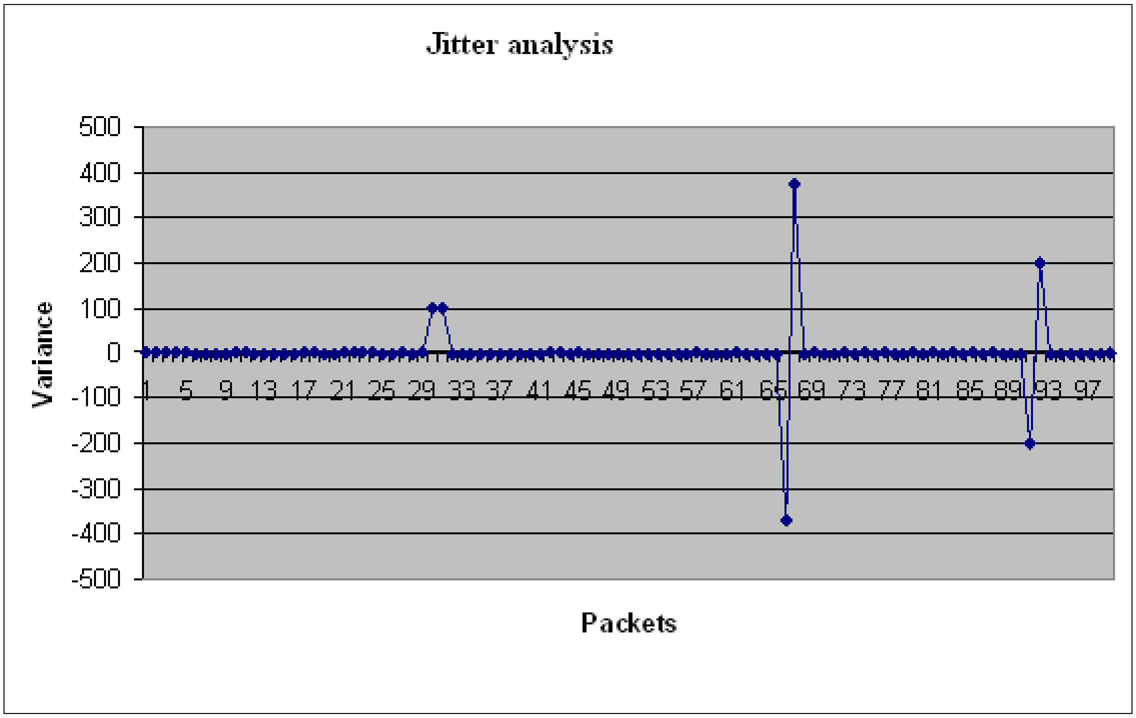


Figure 4.3

2- At data rate of 840,000 bps:

It was observed that when the data traffic was increased to the level of 840000 bps, variation in jitter was also increased. But at this level, it did not affect the video streaming performance. Following figure presents the jitter variation at data rate of 840 kbps.

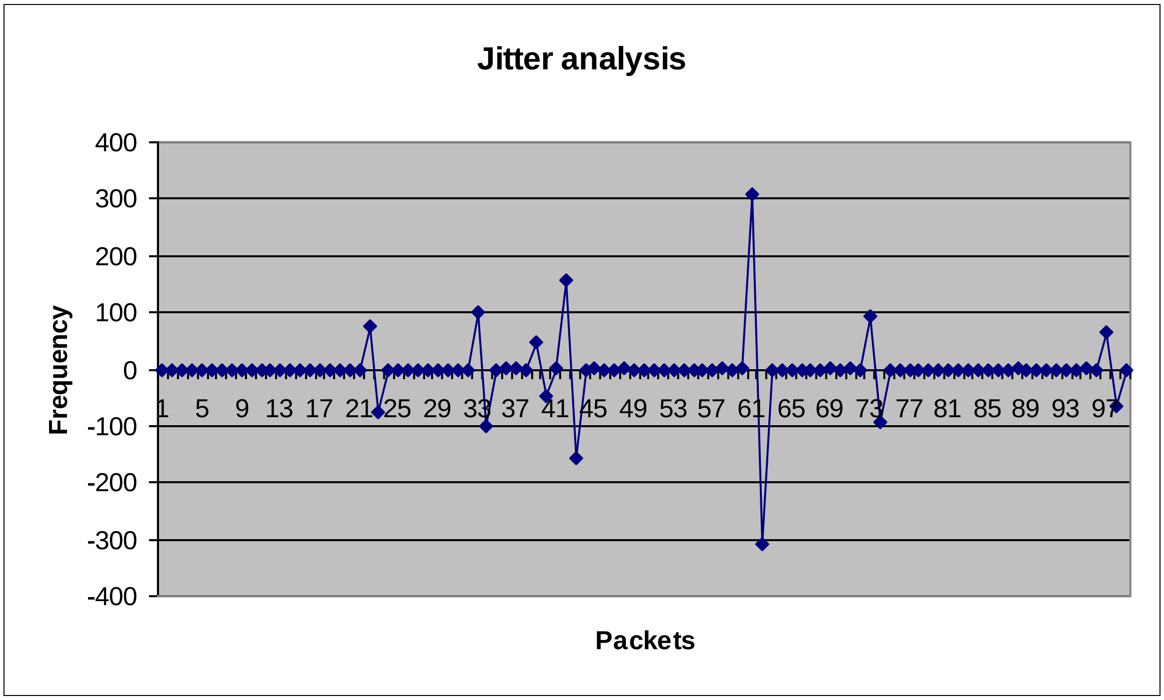


Figure 4.4

3- At data rate of 1 Mbps (threshold level):

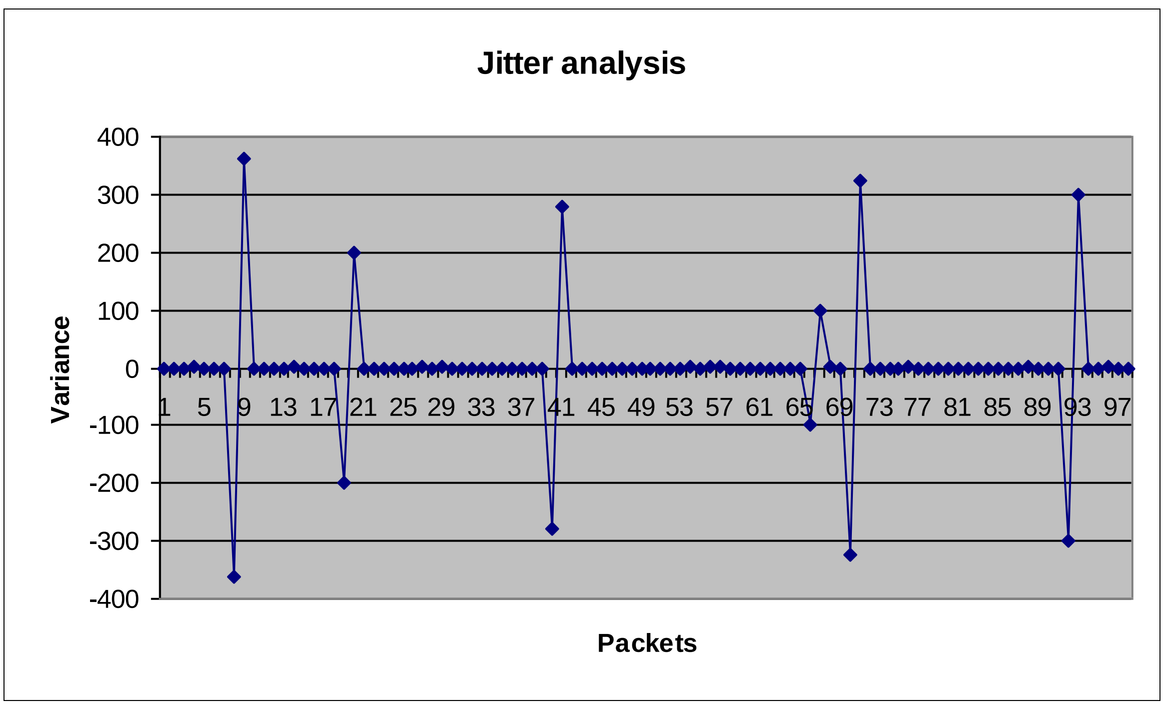


Figure 4.5

This figure gives the behavior of jitter when the traffic is increased to the level of total link capacity. It was found that at this point, the variation in jitter was increased according to the increase in competing data rate. At threshold level, the video quality is maintained because the link has enough space to accommodate the video traffic.

4- At data rate of 1,024,000 bps (above the link capacity):

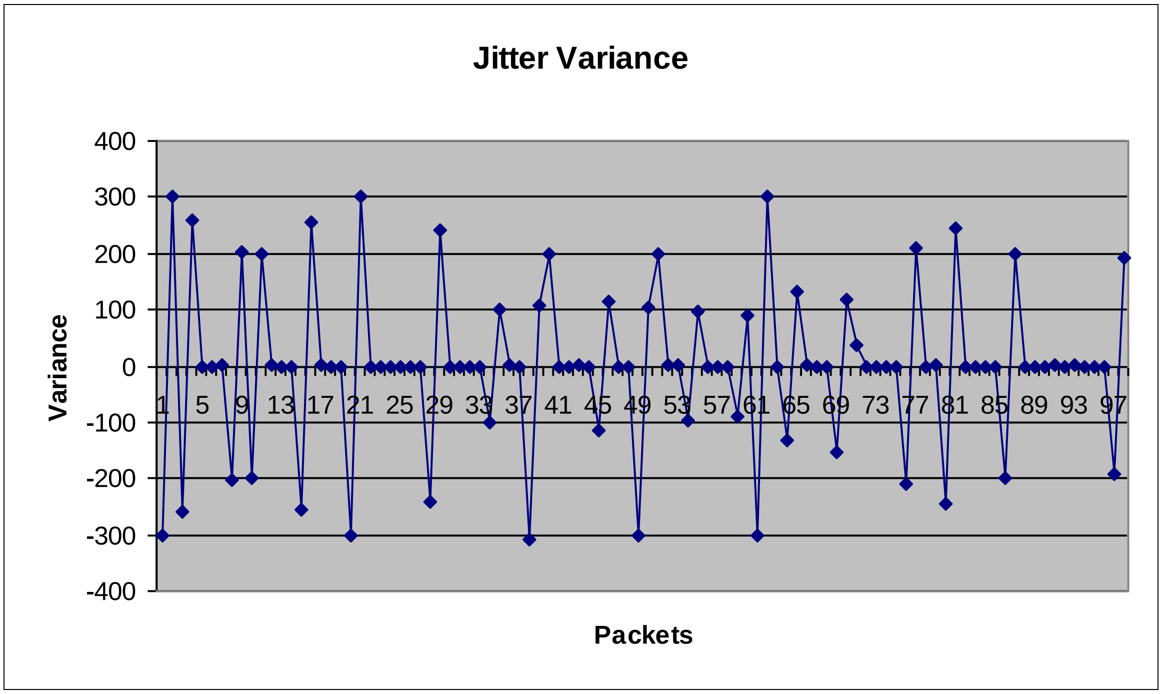


Figure 4.6

This figure shows a high variation in jitter. It was found that when the traffic was increased the link capacity, a high variation in jitter was observed and client could not be able to play back the video contents properly, meaning the video became frozen.

4.3-Throughput

Competing data traffic has a major impact on the throughput of the video data traffic. As the competing data traffic is increased, the throughput of video data is decreased and vise versa, because the available link is now consumed by two types of traffic (video traffic + other data traffic).

Note: HTTP and RTSP-TCP behaves in same fashion. So, TCP represents both the HTTP and RTSP-TCP in the figure below. And UDP represents the RTSP-UDP.

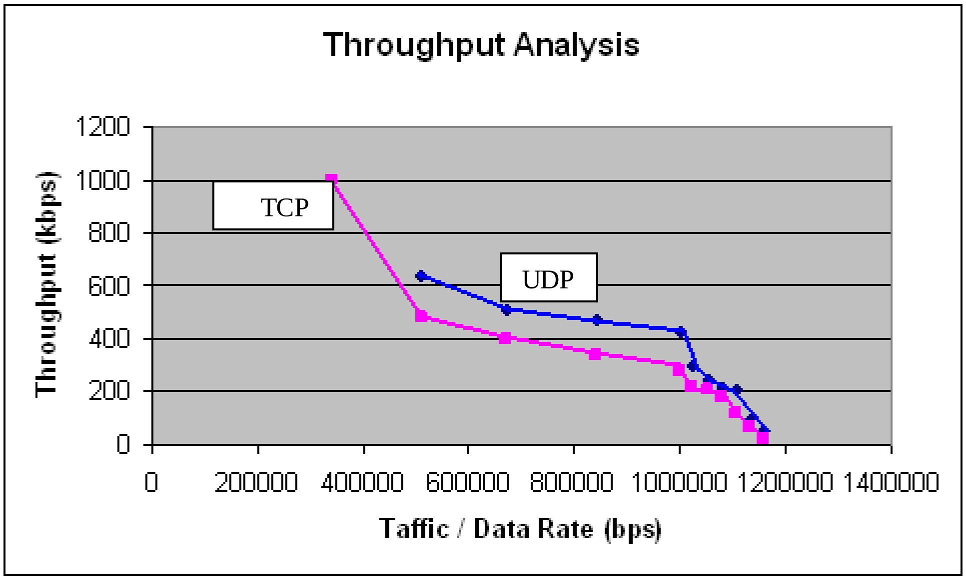


Figure 4.7

This figure presents the relationship between throughput and traffic rates. The throughput was measured at different data rates and the obtained results shows that as the competing data traffic increases, the required throughput for video traffic decreases, as shown in the figure.

4.4-Packet drop and Frame rate

Competing data traffic has a major impact on frame rate. When the total traffic rate was kept below or equals the link speed, the frame rate remained close to 30 fps. But as the total traffic rate exceeded the threshold level, a significant decrease in frame rate was observed. As a result, a huge packet drop was observed at the router serial port; due to which client could not be able to play back the video properly. Ultimately a huge reduction in the frame rate was noticed. This behavior can be seen in the figure below.

Note: Note: HTTP and RTSP-TCP behaves in same fashion. So, TCP represents both the HTTP and RTSP-TCP in the figure below. And UDP represents the RTSP-UDP.

Frame rate behavior at different data rates is given below.

In case of TCP:

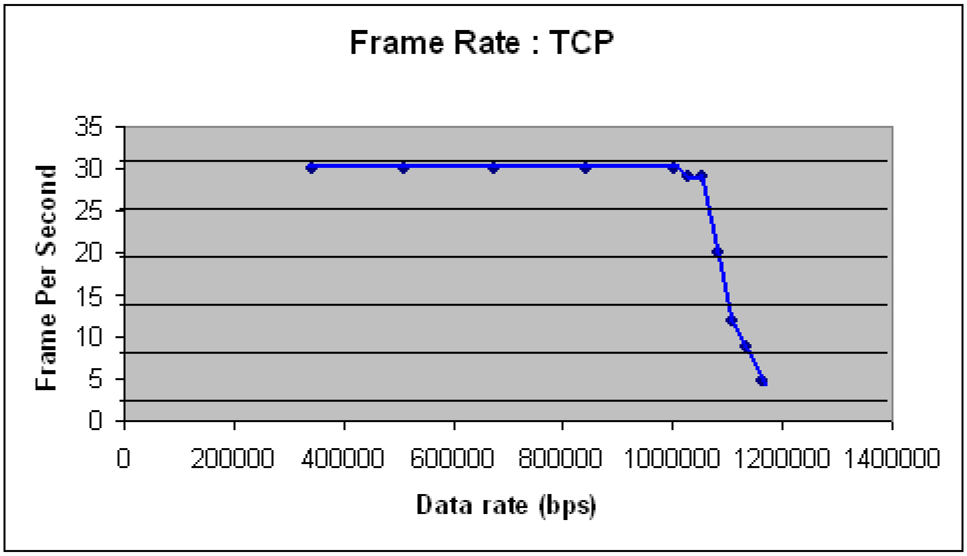


Figure 4.8

In case of UDP:

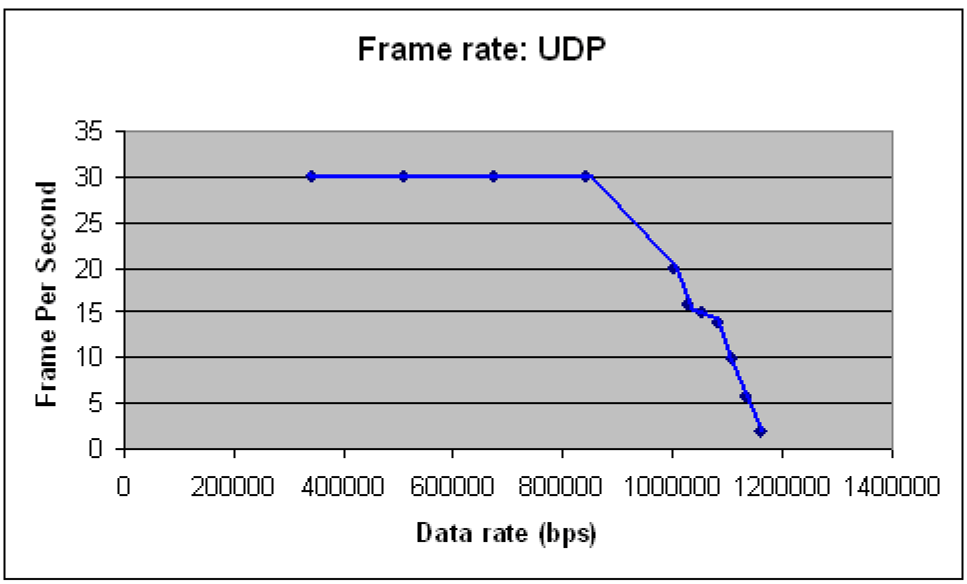


Figure 4.9

The figure 4.8 and 4.9 shows that as the traffic rate was increased, the frame rate was decreased and vise versa.

4.5- Router CPU utilization

Router CPU utilization was also changed in the presence of the competing data traffic. As the data traffic is increased, the router CPU utilization was also increased. Up to 20 % of CPU utilization was observed in case of UDP and 9% CPU utilization was observed in case of TCP, when the total traffic was kept above the link capacity. CPU utilization remained very low (up to 6%), when the data traffic was kept equal or below the threshold level (1 Mbps i.e. total link capacity) as shown in figure 4.10.

Note: Note: HTTP and RTSP-TCP behaves in same fashion. So, TCP represents both the HTTP and RTSP-TCP in the figure below. And UDP represents the RTSP-UDP.

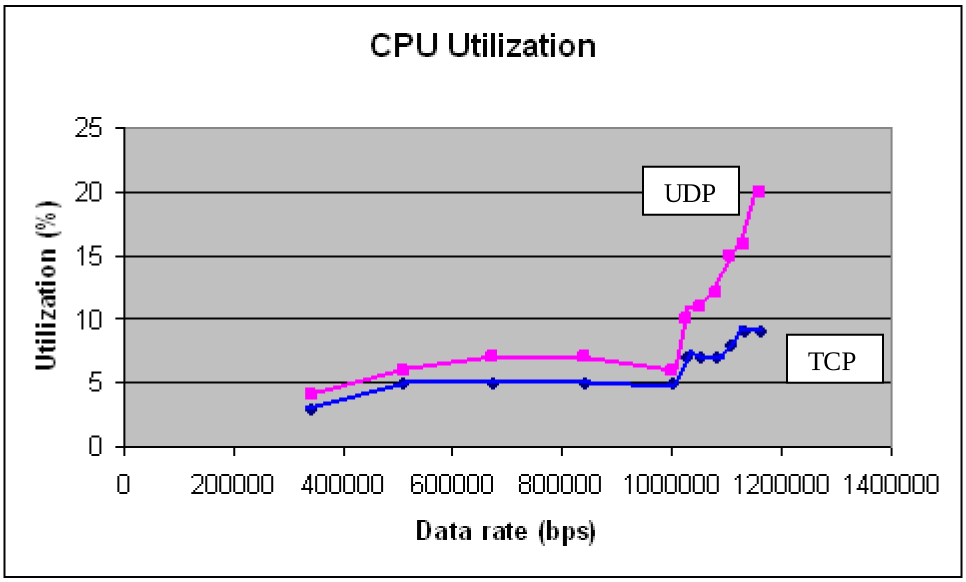


Figure 4.10

4.6 Summary

Delay value is increased as the rate of competing data traffic is increased. When the traffic rate exceeds the total link speed, a very high delay is observed and due to which client could not play back the video streams properly

The high variation in interarrival time was observed, when the video traffic is streamed along with other competing data traffic. And it keeps increasing as the competing data traffic rate is increased. As the competing data traffic is increased, the throughput of video data is decreased.

Increase in packet loss is noticed when the competing data traffic is introduced. This ultimately decreases the frame rate. As the data traffic is increased, the router CPU utilization is also increased.

Chapter-5

Implementing Priority Queuing (PQ) mechanism while transferring the video contents along with other competing data traffic.

For proper video display on client side, the video data should arrive in a smooth and continuous stream. In a crowded network (network with different types of traffic running on), it is hard to achieve the video quality properly. One way to achieve the proper video quality in the busy network is to implement the priority queuing mechanism. In priority queuing, different queues are assigned to different types of data according to their importance. This instructs the route or other network devices to give priority to the special data traffic.

In our project, video traffic is most important that other competing data traffic. So, we gave the high priority to the queue handling the video traffic and low priority to the queue handling the other data traffic. This mechanism was implemented over router Ethernet port (port receiving both the video traffic and other data traffic). As a result, the router gave high priority to video data over other competing traffics, and transferred the video traffic with any further delay.

The effect of priority queuing was studied in terms of various performance issues and it was found that this mechanism helped to achieve the proper video quality in the presence of other competing data traffics.

In this case, 1 Mbps link was used. Video data rate was 380 bps. In this case, we can send another 620 bps of data over this link without affecting the video traffic performance, and there is not need of priority queuing. But when the incoming traffic is increased than the link capacity (1Mbps), it starts dropping the video data packets along with other competing data traffic. As a result, the video quality is degraded and clients receive the video contents with poor quality. The video quality keeps degrading as the competing data traffic is increased and vise versa. So at this point we need some mechanism that can save the video quality and make video streaming reliable. So, priority mechanism was used to achieve reliable video streaming, Video performance issues were analyzed in the presence of competing data traffic while implementing the priority queuing mechanism.

5.1-Delay

Priority queuing has an important impact on delay. While implementing PQ, it was found that delay remained constant as the data traffic is increased than the threshold level. A very little increase in delay was noticed. The three protocols, HTTP, RTSP-TCP and RTSP-UDP behaved in the same fashion in the presence of PQ. This behavior is shown below in the figure.

In case of actual delay (excluding the queuing delay) a very constant delay was observed. This constant delay value is achieved due to priority queuing mechanism. The figures below show that PQ controls the variation in delay by assigning a special queue to the video traffic, so the video contents can be transferred without any further delay.

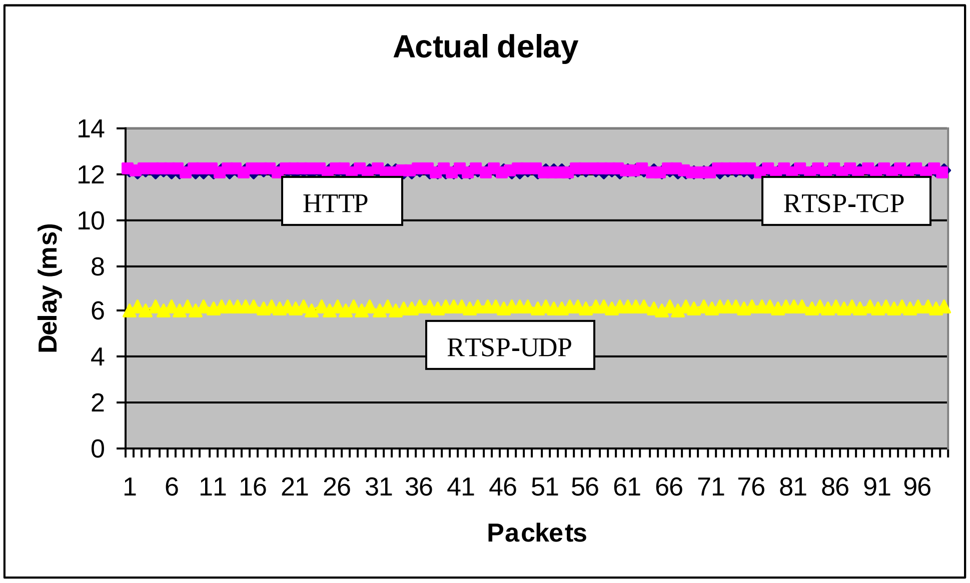


Figure 5.1

In case of queuing delay, delay increased in a constant way without degrading the video quality. This constant increase is shown below.

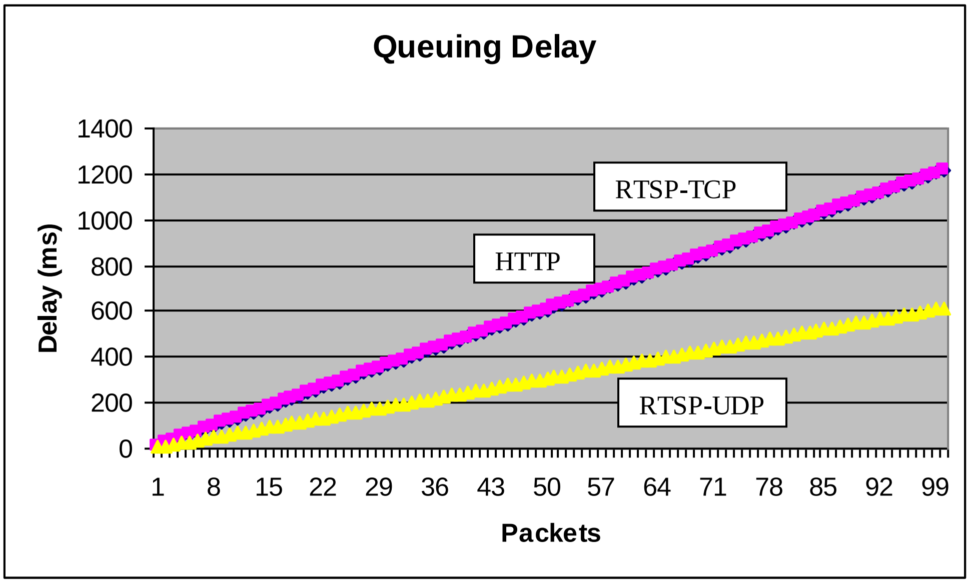


Figure 5.2

Note: Queuing delay + Actual delay = Total delay

5.2-Jitter

Jitter is another factor that can be influenced by PQ mechanism. A very small variation in jitter was observed while transferring the video contents in the presence of other competing traffic while implementing the PQ. UDP and TCP responded in the same fashion. This behavior can be seen in figure 5.3 and 5.4.

Note: HTTP and RTSP-TCP behaves in same fashion. So, TCP represents both the HTTP and RTSP-TCP in the figure below. And UDP represents the RTSP-UDP.

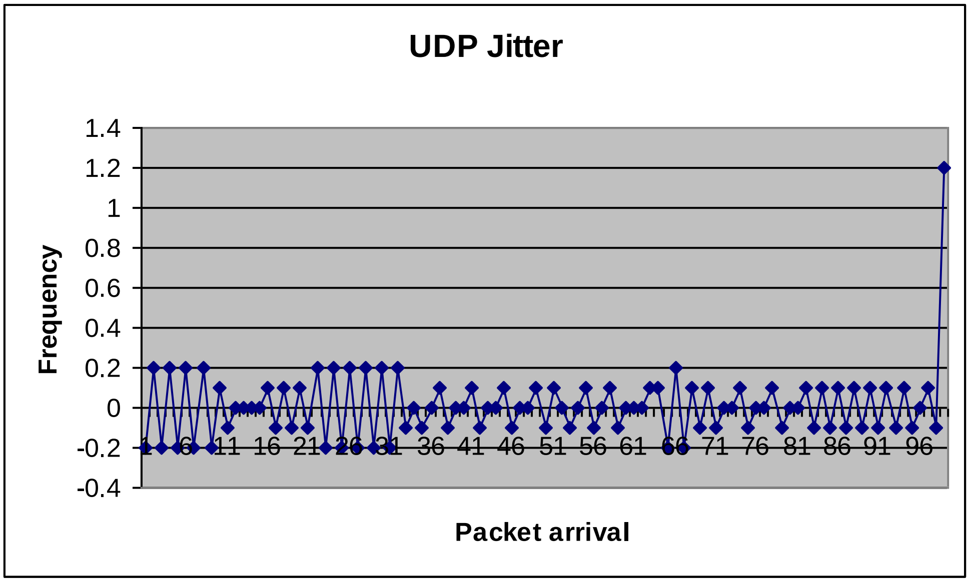


Figure 5.3

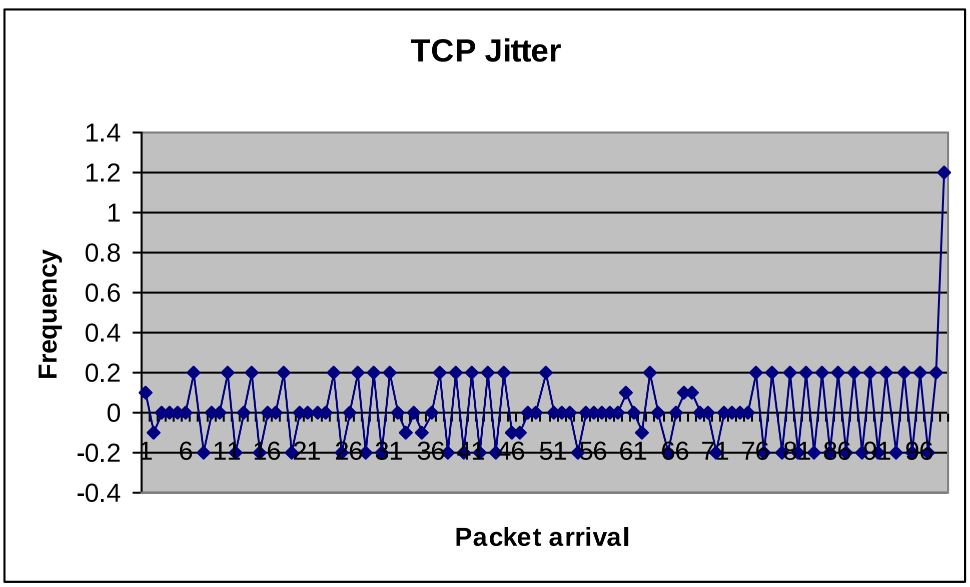


Figure 5.4

5.3- Throughput

By implementing priority queuing, a router can allocate the link bandwidth according to the required video data rate. In this project, the video with data rate of 308 kbps is used.

It was found that as the competing data traffic was increased over the threshold level, the video quality was not degraded because the router assigned the 390 kbps to 460 kbps of bandwidth to the video traffic on the basis of the total traffic rate. Therefore, the client received the good quality video contents. This achieved throughput at various traffic rates is presented in the following figure.

Note: HTTP and RTSP-TCP behaves in same fashion. So, TCP represents both the HTTP and RTSP-TCP in the figure below. And UDP represents the RTSP-UDP.

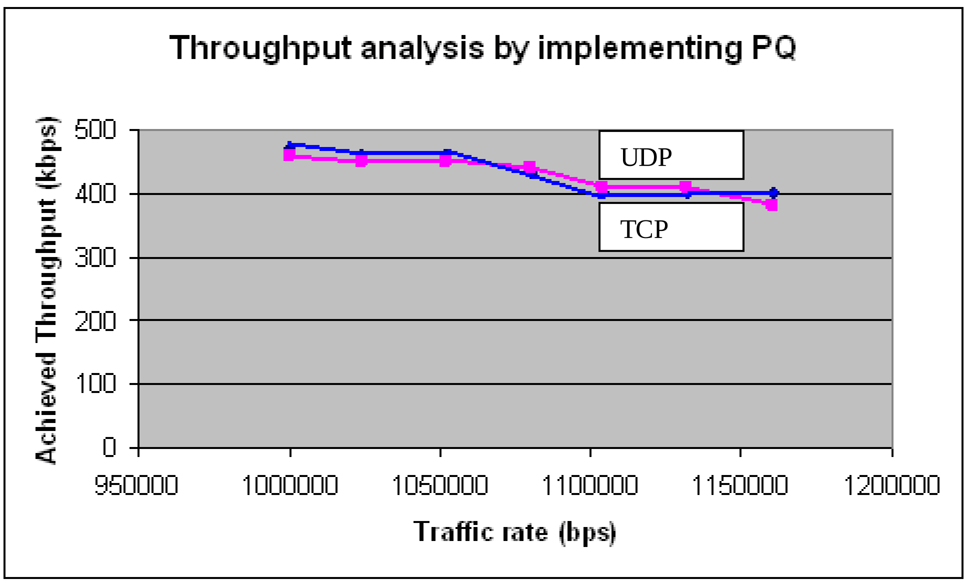


Figure 5.5

The figure shows that the throughput remained close to the bandwidth required to transfer the video contents properly (i.e. 340 kbps).

5.4-Packet drop and Frame rate

As the competing traffic was increased than the available link capacity, the packet loss occurred and ultimately the frame rate was dropped (as shown in case2 of the project). But when we implemented PQ mechanism, the packet drop was stopped. A very little video packet drop was observed as the traffic kept increasing the link capacity. At this stage, the competing data packets were dropped instead of video data packets. And ultimately, video with high frame rate (up to 30 fps) was achieved at client side. This all was done due to PQ.

This behavior is shown below: All three protocols behave in the same fashion.

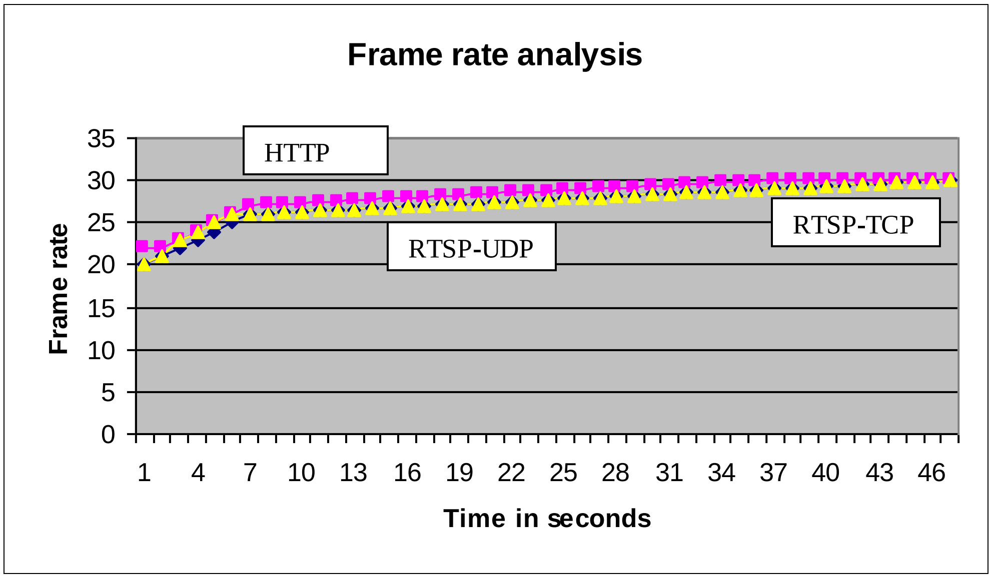


Figure 5.6

5.5-Router CPU-utilization

PQ has a vital effect over router CPU-utilization. As the data rate is increased the router CPU- utilization tends to increase. Because the router has to do some extra work in prioritizing the traffic and then forwarding the traffic accordingly. The UDP (RTSP-UDP) and HTTP (RTSP-TCP and HTTP) behaves in same way.

It was found that when, the competing data traffic was increased than the link capacity, and the router CPU utilization was kept increasing and vise versa. And utilization was reached at the level of 36 %. This behavior is shown in the figure below:

Note: HTTP and RTSP-TCP behaves in same fashion. So, TCP represents both the HTTP and RTSP-TCP in the figure below. And UDP represents the RTSP-UDP.
Router CPU- behavior is shown below:

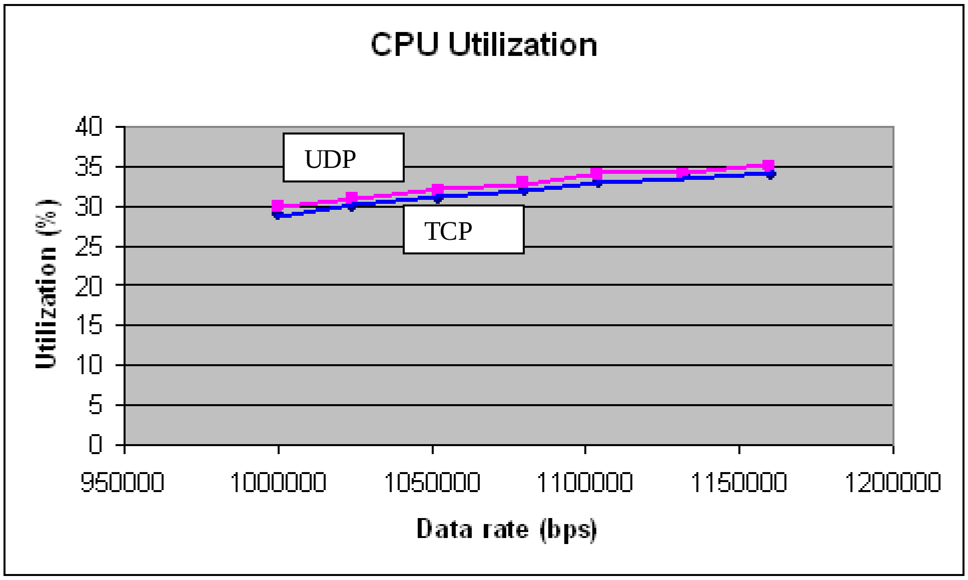


Figure 5.7

5.6 Summary

While implementing PQ, it was found that delay remains constant as the data traffic is increased over the threshold level. A very small variation in jitter was observed while transferring the video contents in the presence of other competing traffic while implementing the PQ.

By implementing priority queuing, a router can allocate the link bandwidth according to the required video data rate. PQ mechanism stops the packet drop as the video contents are streamed on priority basis over the crowded network. And ultimately a high frame rate (up to 30 fps) is achieved.

As the data rate increases, the router CPU- utilization also increases. Because the router has to do some extra work in prioritizing the traffic and then forwarding the traffic accordingly.

5.7 Summary tables

Comparison of video performance under nominal and loaded conditions, with and without PQ

	HTTP	RTSP-TCP	RTSP-UDP
Under normal conditions	Excellent video quality	Excellent video quality	Excellent video quality
Under loaded conditions at various traffic rates without PQ			
1-Equal or below the link rate i.e. 1 Mbps	Excellent video quality	Excellent video quality	Good video quality Somewhat blocky at beginning as the client connects to video stream
2-Above the link rate at 1,024,000 bps	Blocky	Blocky	Blocky
At 1,052,000 bps	Blocky	Blocky	Blocky
At 1,104,000 bps	Video froze	Video froze	Video froze
At 1,132,000 bps	Video froze	Video froze	Video froze
Under loaded conditions (above the link rate) with PQ	Excellent	Excellent	Excellent

Figure 5.8

Actual Delay and Jitter Analysis

	HTTP	**RTSP-TCP**	**RTSP-UDP**
Under normal conditions	Remains constant	Remains constant	Remains constant
Under loaded conditions without PQ			
1-Traffic equal or below the link rate	Keep increasing to a certain level (as the competing data traffic increases) that doesn't degrade video quality	Keep increasing to a certain level (as the competing data traffic increases) that doesn't degrade video quality	Keep increasing to a certain level (as the competing data traffic increases) that doesn't degrade video quality
2-Traffic above the link rate	Keep increasing as the competing data traffic increases and this degrades video quality (blocky + froze)	Keep increasing as the competing data traffic increases and this degrades video quality (blocky + froze)	Keep increasing as the competing data traffic increases and this degrades video quality (blocky + froze)
Under loaded conditions with PQ	Remains constant	Remains constant	Remains constant

Figure 5.9

Note:

Actual delay = Total delay – Queuing delay

Packet loss / Frame rate analysis

	HTTP	RTSP-TCP	RTSP-UDP
Under normal conditions	0 packet loss 29-30 fps	0 packet loss 29-30 fps	0 packet loss 29-30 fps
Under loaded conditions at various traffic rates without PQ			
1-At 672,000 bps	0 packet loss 29-30 fps	0 packet loss 29-30 fps	0 packet loss 26-30 fps
2-At 840,000 bps	0 packet loss 29 fps	0 packet loss 29 fps	0 packet loss 25-29 fps
3-At 1 Mbps (link capacity)	0 packet loss 29 fps	0 packet loss 29 fps	20-22 fps a small packet loss was observed at beginning as the client connects to video server
4-At 1,024,000 bps	24 fps with up to 5 % packet loss	23-24 fps with up to 5 % packet loss	15-16 fps with up to 40 % packet loss
5-At 1,052,000 bps	21-22 fps with up to 20 % packet loss	21-22 fps with up to 20 % packet loss	10-13 fps with up to 60 % packet loss
6-At 1,104,000 bps	7-12 fps with up to 60 % packet loss	7-12 fps with up to 60 % packet loss	5-6 fps with up to 75 % packet loss
7-At 1,132,000 bps	4-5 fps with huge packet loss	4-5 fps with huge packet drop	1-3 fps with huge packet drop
Under loaded conditions (above the link rate) with PQ	Up to 30 fps with no packet loss	Up to 30 fps with no packet loss	Up to 30 fps with no packet loss

Figure 5.10

Throughput analysis

	HTTP	**RTSP-TCP**	**RTSP-UDP**
Under normal conditions	Max. throughput utilization i.e.1Mbps	1 Mbps	1Mbps
Under loaded conditions at various traffic rates without PQ			
1-At 672,000 bps	400 kbps Good video quality	400 kbps Good quality	510 kbps Good quality
2-At 840,000 bps	380 kbps Good quality	380 kbps Good quality	470 kbps Good quality
3-At 1 Mbps (link capacity)	370 kbps; enough to stream video properly	370 kbps; enough to stream video properly	380 kbps; enough to stream video properly
4-At 1,024,000 bps	200 kbps Blocky	200 kbps Blocky	210 kbps Blocky
5-At 1052,000 bps	180 kbps Blocky	180 kbps Blocky	200 kbps Blocky
6-At 1,104,000 bps	120 kbps Video froze	120 kbps Video froze	140 kbps Video froze
7-At 1,132,000 bps	90 kbps Video froze	90 kbps Video froze	110 kbps Video froze
Under loaded conditions (above the link rate) with PQ	400-470 kbps; enough to stream video properly	400-470 kbps; enough to stream video properly	380-460 kbps; enough to stream video properly

Figure 5.11

Router CPU-Utilization analysis

	HTTP	**RTSP-TCP**	**RTSP-UDP**
Under normal conditions	2-3 %	2-3 %	2-3 %
Under loaded conditions without PQ			
1-Traffic equal or below the link rate	5 %	5 %	6 %
2-Traffic above the link rate	9 %	9 %	20 %
Under loaded conditions (above the link rate) with PQ	30-36 %	30-36 %	30-36 %

Figure 5.12

Chapter-6

Conclusions and future directions

This project was carried out to study the performance of video streaming in terms of delay, jitter, throughput utilization, packet loss, frame rate and router CPU-utilization using the Windows Media Player. It was found that when there is not other competing data traffic, the achieved video quality remains reliable and high.

In case of competing data traffic, the video quality is degraded as the competing data traffic is increased.

The adverse effect of competing data traffic on video streaming could be resolved by implementing the priority queuing mechanism. PQ gives priority to the video traffic over all other competing traffic, by assigning it a separate queue with high priority and hence maintaining the video quality.

This study could be helpful in analyzing the performance of video communication in the crowded network. PQ could be implemented in websites and organizations where video traffic is more important than other data traffics.

.Future Directions

This information would be helpful to study the video performance analysis studies by using other commercially used media players where we give the highest priority to the video traffic over all other competing data traffic.

The information will also be useful for researchers, video service provider and vendors to compare the video streaming performance of Windows Media Player with other video streaming players and to decide their suitability and performance capacity.

This study would also be helpful to investigate the limitations of PQ as the number of video streams increases. Knowing at what point PQ fails is important to service providers like TELUS.

Chapter-7

References:

[1] J. G. Apostolopoulos, W. Tan and S. J. Wee, "Video Streaming: Concepts, Algorithms and Systems," Mobile and Media Systems Laboratory, HP Laboratories, Palo Alto HPL-2002-260, Sept. 2002.

[2] Y. Wang, M. Claypool and Z. Zuo, "An empirical study of RealVideo Performance across the Internet."

[3] V. Mariappan and P. Narayanasamy, "Media Streaming over WLAN using Stream Control Transmission Protocol," Dept. of Computer Science & Engineering, Anna University, Chennai, India -600 025.

[4] A. Mena and J. Heidemann. An Empirical Study of Real Audio Traffic. In *Proceedings of the IEEE Infocom*, pp. 101-110. Tel-Aviv, Israel. March 2000.

[5] M. Chesire, A. Wolman, G. Voelker, and H. Levy. Measurement and Analysis of a Streaming Media Workload, In *Proceedings of the USENIX Symposium on Internet Technologies and Systems (USITS)*, San Francisco, CA, USA, March 2001.

[6] Microsoft windows media. [Online]. Available: http://microsoft.com/windows/windowsmedia

[7] Realone player. [Online]. Available: http://www.real.com/realoneplayer.html

[8] Quicktime. [Online]. Available: http://www.apple.com/quicktime

[9] http://www.microsoft.com/windows/windowsmedia/forpros/server/server.aspx

[10] Adapted from "Windows Media Services 9 Series guide.'' Windows\help\wmserver.chm::/htm/comparingstreaminganddownloading.htm

[11] Adapted from "Windows Media Services 9 Series guide.'' Windows\help\wmserver.chm::/htm/protocoloverview.htm

[12] Adapted from “Windows Media Services 9 Series guide.’’
Windows\help\wmserver.chm::/htm/howprotocolrolloverworks.htm

[13] Adapted from “Windows Media Services 9 Series guide.’’
Windows\help\wmserver.chm::/htm/wmshttpservercontrolprotocol.htm

[14] Adapted from “Windows Media Services 9 Series guide.’’
Windows\help\wmserver.chm::/htm/wmsrtspclientcontrolprotocol.htm

[15] Adapted from “Windows Media Services 9 Series guide.’’
Windows\help\wmserver.chm::/htm/wmsmmsservercontrolprotocol.htm