University of Alberta

Master of Science in Internetworking Capstone Project

Report

Video Codecs simulation and compare QoS characteristics of different video codecs.

Ankit Rana
arana@ualberta.ca

Submitted to
Mr. Pete Nanda.

February 17, 2014

Abstract

In this project QoS characteristics of different video codecs are compared. QoS characteristics of H.265 and H.264 video codecs are tested and compared on identical network conditions and load. Traffic is running on a subset sample of Telus network. To carry out the simulation OMNET++ is used, OMNET++ is a network simulating software. The network is flooded with different kinds of traffic like web browsing, voice and video streaming. Behaviour of network is recorded and analysed in different condition like with Qos, without Qos, with congestion and without congestion. In all tests, video streaming is the only changing parameter and other parameters are kept constant.

The purpose of this project is also to find out if OMNET++ can help in learning about the general aspects of computer networking. For example, the devices, the physical and logical aspects of a computer network and topologies.

Table of Contents

1

Introduction

Motivation and Objective: Internet traffic is growing every day, IP networks were not designed for such large quantities of data. There are challenges in delivering data over IP networks. With this project I have an opportunity to understand some of these challenges and work on them. Main area of my study is video traffic over internet. Delivering video is very challenging and with increasing demand service provider are focusing on video coding techniques, by using the latest video codecs they can reduce the bandwidth usage by 50%.

Around 70% of video traffic uses H.264 as a standard video codec, but with recent development in video codecs, the same traffic can be delivered with H.265 video codec using half of network bandwidth. It's a good area of study and can help in understanding the concept of video codecs, network traffic, protocols and network topologies. The project deals with Qos comparison of different video codecs like MPEG 4, H.264, H.265.

For network simulations I am using OMNET++. OMNET++ is a simulation software that predicts the behavior of a computer network without an actual network being present. In simulators, the computer network is typically modeled with devices, traffic etc. and the performance is analysed.

The final goal of the project:

1. Understanding video codec, video streaming using network simulation software.
2. Comparison of QoS characteristics of different video codecs.
3. Behavior of network dealing with video traffic.
4. Understand the network behavior against certain protocol messages.
5. Understand the simulation software OMNET++.

2

Challenges in video-over-IP transmission

While video represent a great opportunity and traffic is growing day by day. IP networks fundamentally were not designed for such large quantities of real time sensitive video content. As we can imagine, there are challenges in delivering streaming video over IP networks. Beside design of IP network there lot more video delivery challenges. Below are some key challenges faced by service providers.

- **Demanding traffic profiles:** Video traffic is real time with strict delay, loss requirement and delay variation. High bandwidth streams further stress the network's ability to deliver video in high quality.

- **Video standards:** If we are sending a type of video that is not supported by the receiving equipment, then obviously the system will not work.
- **Compression standards:** As with video standards, it's not a good idea to use incompatible compression standards between the sender and the receiver. Feeding MPEG4 into an MPEG-2 decoder will not achieve acceptable results. Achieving interoperability across different MPEG2 systems — while much better than in the past — may occasionally cause problems. If we are building one of these systems, be sure to test compatibility between differing encoder and decoder combinations.
- **High End-User Expectations:** Internet protocols, having built primarily with scalability and fairness in mind, do not perform well with large volumes of real-time video traffic. Packet loss, designed into IP Networks to maintain overall performance objectives, has adverse affects on video.

- **FEC standards:** It is important to use standardized FEC if we want to ensure that error correction capabilities function correctly when interoperating between different equipment manufacturers.

If we are designing a system that transports video over IP within a single company from one point to another, then interoperability is a nonissue. We decide on the video standards, compression standards, forward error correction (FEC) method and Internet standards.

On the other hand, if we are working with a system that transports video between different entities, or even within different branches of the same entity that use different corporate standards, then we may face some challenges. Of course, the issue of interoperability is not new, but video over IP adds a few new parameters. Overall, there are standards exist to resolve these interoperability issues. Manufacturers and service providers are working together to ensure that video-over-IP installations go smoothly.

Different methods of delivering Video over IP networks

At the Transport layer of the IP network stack, UDP (User Datagram Protocol) is the preferred method for the delivery of live video streams. UDP offers reduced latency over the reliability that TCP (Transmission Control Protocol) provides. It is a faster protocol than TCP and where time sensitive applications are involved (i.e. live video or VoIP), it is better to live with a video glitch caused by a dropped packet than to wait for the retransmission which TCP guarantees (which in not very practical where live video is concerned).

The Internet now has a global reach and so has become a suitable medium to distribute multimedia content, and providers use various methods to deliver video like Unicast and Multicast

Multicast is a method of one-to-many transmission which is often deployed in IP applications of streaming media. Multiple viewers can simultaneously tap into a single transmission from one source. It is often employed for streaming media applications on the Internet and private networks. Each host (and in fact each application on the host) that wants to be a receiving member of a multicast group must use the Internet Group Management Protocol (IGMP) to join. IGMP operates between the client computer and a local multicast router. Protocol Independent Multicast (PIM) is then used between the local and remote multicast routers, to direct multicast traffic from the multicast server to many multicast clients.

With Unicast transmission, every user in the network who would like to view video will receive a dedicated video stream from the Video Management System. Compared to Multicast transmission, Unicast does utilize more bandwidth; however, these streams are only required between the source and the viewer, and do not affect the entire network

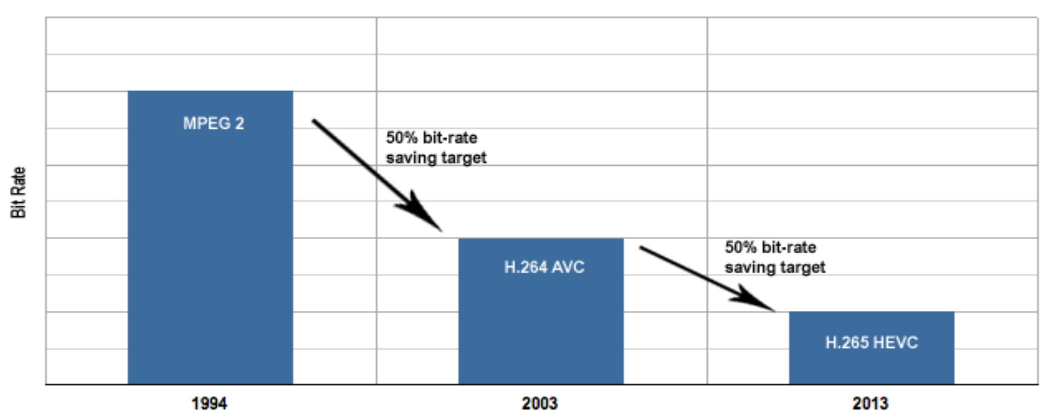
Multicast transmissions do offer the benefit of lower bandwidth consumption, but comes with higher network construction cost. Unicast provide cost-savings on the construction of Multicast-enabled networks while maintaining real-time, low latency, high-quality video with adequate bandwidth management for all users on the network.

3

HEVC Video Codec feature enhancements over its predecessors

H.264/AVC (Advanced Video Coding) is currently the most commonly used formats for the recording, compression, and distribution of high definition video. H.264 encoding is responsible for delivering the video on web, It is widely used by streaming internet sources, such as videos from Dailymotion, YouTube, and the iTunes Store, about 80% of web video now runs on H.264 codec. But still it is not yet ready to encode very high resolution videos like 4k and 8k videos, which is why a new standard is needed which can meet the demand.

The solution to this problem is H.265/HEVC (High Efficiency Video Coding**)**, it is a successor to H.264/MPEG-4 AVC. HEVC double the data compression ratio as compared to AVC at the same level of video quality. HEVC will support video files of 4K and 8K resolutions while simultaneously improving upon current streaming by cutting the required bitrate by up to 50 percent. The way it does this is relying on more specialized hardware and computing power, with improved compression algorithms and more time to a computer to process a video, the more compact a video can be compressed without losing too much data. H.265 will require more computational power than its predecessor, as computer and mobile CPU's power is increasing every year as compared to network infrastructure it is now possible to use H.265 as main codec on web.



Bit players: each successive generation of video codec delivers comparable picture quality at half its predecessor's bit-rate

HEVC was designed to improve coding efficiency compared to H.264/AVC, to reduce bitrate requirements half with comparable image quality. Key features where HEVC is improved compared to H.264/AVC are as follow.

- The core of the coding layer in previous standards was the macroblock, containing a 16×16 block of luma samples and two corresponding 8×8 blocks of chroma samples, whereas the structure in HEVC is the coding tree unit (CTU), which has a size selected by the encoder and can be larger than a traditional macroblock. It can extend the block size up to 64x64.

- H.265 is designed to use a more efficient means of encoding pixel data and incorporates larger blocks of pixels than H.264 macroblocks did. It can divide a picture into tiles for more efficient parallel processing and decode slices independently for better resynchronization.

- HEVC uses three frame types, I-, B- and P-frames within a group of pictures, incorporating elements of both inter-frame and intra-frame compression.

- HEVC specifies 33 directional modes for intra prediction compared to the 8 directional modes for intra prediction specified by H.264/AVC.

- Adaptive Motion Vector Prediction, which allows the codec to find more inter-frame redundancies.

- Superior parallelization tools, including wave front parallel processing, for more efficient encoding in a multi-core environment.

Long-term, H.265 will likely succeed H.264′s position as the premier solution for advanced video. H.265′s explicitly parallel model should map well against multi-core devices of the future. When used well together, the features of the new design provide approximately a 50% bit-rate savings for equivalent quality relative to the performance of prior standard AVC. The complexity of HEVC as compared to H.264/AVC is not a major problem as with the use of modern processing technology, encoder complexity is manageable

4

Design and Implementation

OMNET++ is used for network simulations. OMNeT++ stands for Objective Modular Network Test bed in C++**.** It is a discrete event simulation tool designed to simulate computer networks, multi-processors and other distributed systems. Its applications can be extended for modelling other systems as well. It has become a popular network simulation tool in the scientific community as well as in industry over the years.

Main components of OMNET++:

- Simulation kernel library.
- Compiler for the NED topology description language (nedc).
- Graphical network editor for NED files (GNED).
- GUI for simulation execution, links into simulation executable (Tkenv).
- Command-line user interface for simulation execution (Cmdenv).
- Graphical output vector plotting tool.
- Documentation, sample simulations, contributed material, etc.

Network Model

The network consists of “nodes” connected by “links”. The nodes representing blocks, entities, modules, etc, while the link representing channels, connections, etc. The structure of how fixed elements (i.e nodes) in a network are interconnected together is called topology.

Omnet++ uses NED language, thus allowing for a more user friendly and accessible environment for creation and editing. It can be created with any text-processing tool (perl, awk, etc). It has a human-readable textual topology. It also uses the same format as that of a graphical editor. It also supports sub-module testing. Omnet++ allows for the creation of a driver entity to build a network at run-time by program.

The network consist of client nodes, server nodes, Ethernet links routers and switches. As per the requirement a simple client server model is used as shown in fig 1. Three different servers are used for three type of traffic video, voice and web server. Ethernet links have a 1Gbps, 100Mbps bandwidth and is shared by different type of traffic.

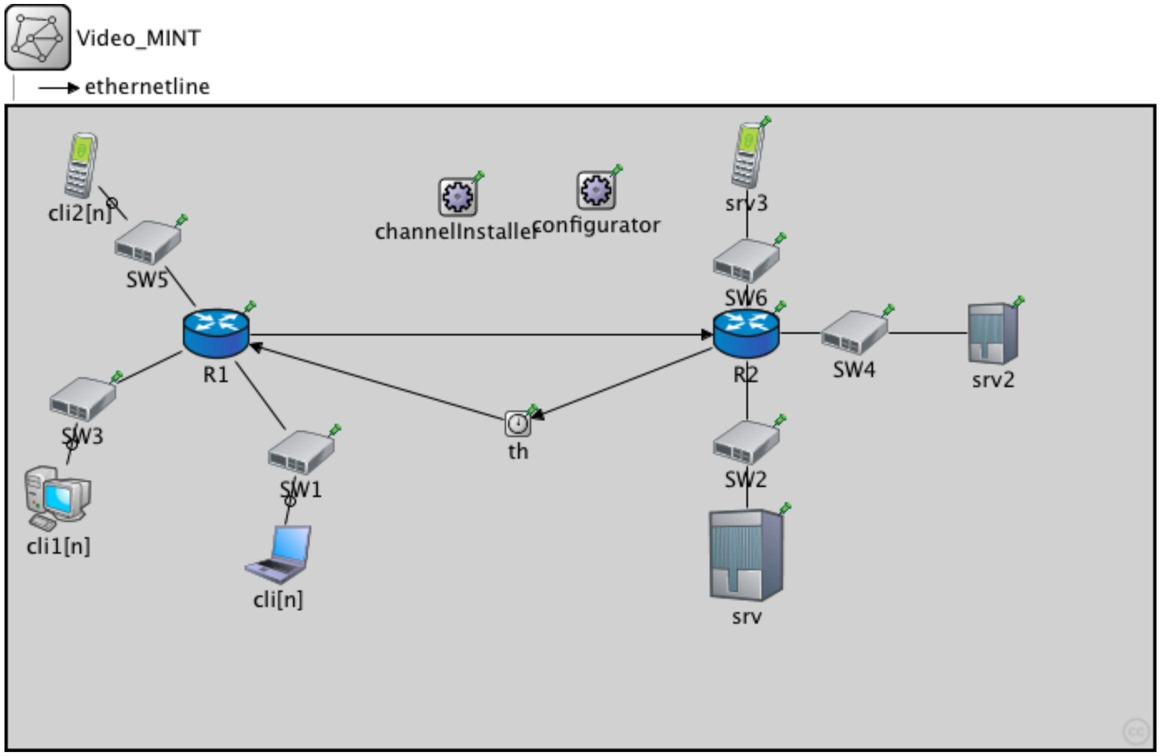


Figure 1: Network topology

Configuration File

In OMNeT++, simulation models are parameterized and configured for execution using configuration files with the .ini extension, called INI files. The INI file contains the configuration of simulation runs. The content of the INI file is divided into sections. In the simplest case, all parameters are set in the General section.

On the Parameters page of the form editor, I set module parameters. First, I have to select the section where the parameters are stored. After selecting the section from the list, the form shows the name of the edited network and the fallback section.

The INI file contain parameter for video, voice and web browsing:

1. **Video**: The Video Interface model in the OMNeT++ implements a trace-based model of video traffic. Traces for all video codecs are supported by this module, and used with INET-HNRL. The maximum payload size is 1472 bytes.

2. **Voice**: Modeling voice traffic as a CBR stream suffices, Traffic generator modules in the inet.applications, packages like UDPBasicApp, can generate both constant bit-rate and variable bit-rate traffic. The packet length is 1460 bytes. Usually the packet size for

VoIP packet is 64-200 Bytes, but in this simulation to congest the links the packet size is changed to 1460 Bytes to increase the traffic.

3. **Web Browsing:** Modules in the inet.applications.tcp package, e.g. TCPSessionApp can be used to model file transfers.

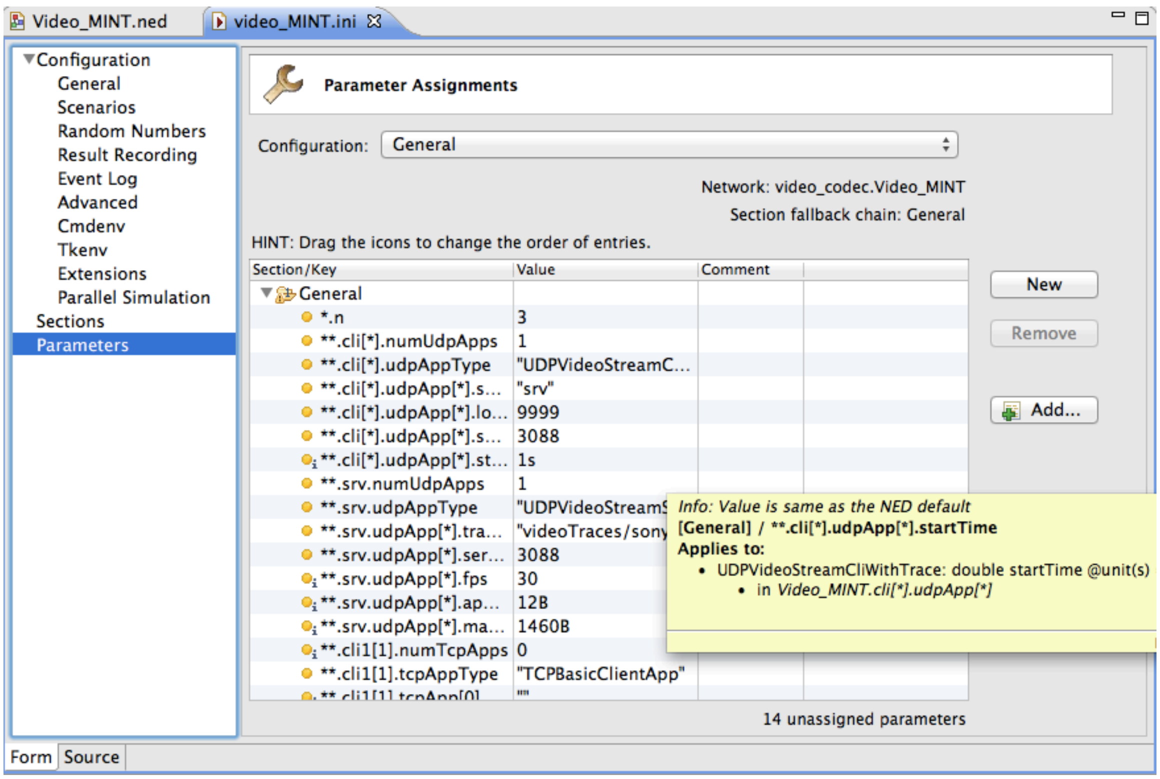


Figure 2: Ini File Parameters

Organization of Network Simulation

Omnet++ follows a hierarchical module structure allowing for different levels of organization.

- **Physical Layer:**

1. Top-level network
2. Sub-network (site)
3. LAN
4. Node

- **Topology within a node:**

1. OSI layers. The Data-Link, Network, Transport, Application layers are of greater importance.
2. Applications/protocols within a layer.

In this simulation the video server is responding to the client request using a unicast, but in real scenario most service provider uses IP multicast. IP multicast is a bandwidth-conserving technology that reduces traffic by simultaneously delivering a single stream of information to potentially thousands of corporate recipients and homes. IP multicast delivers application source traffic to multiple receivers without burdening the source or the receivers while using a minimum of network bandwidth. Multicast packets are replicated in the network at the point where paths diverge by routers enabled with Protocol Independent Multicast (PIM) and other supporting multicast protocols like Internet Group Management Protocol (IGMP), High-bandwidth applications, such as MPEG video, may require a large portion of the available network bandwidth for a single stream. In these applications, IP multicast is an efficient way to send to more than one receiver simultaneously.

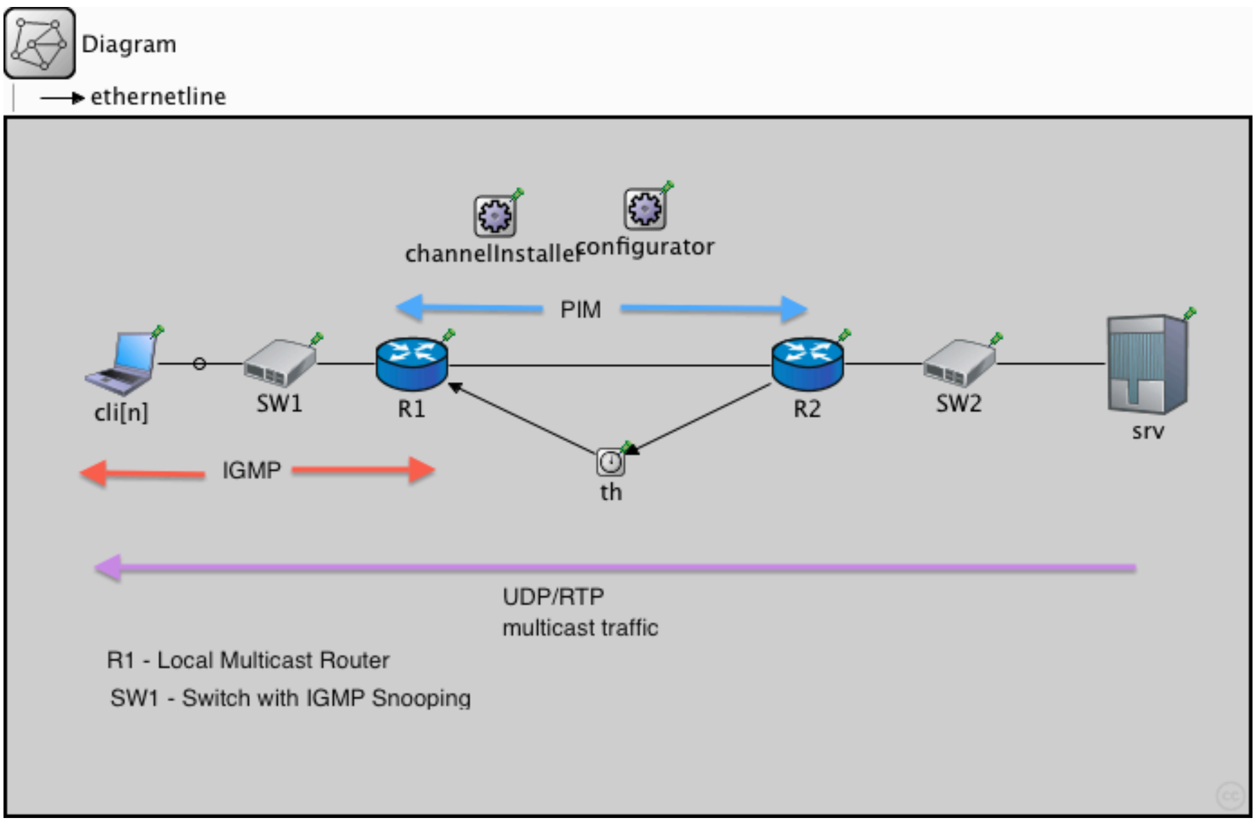


Figure 3: Multicast

IGMP is used by hosts and adjacent routers on IP networks to establish a multicast group memberships. IGMP operates between the client computer (cli[n]) and a local multicast router (R1). Switches featuring IGMP snooping derive useful information by observing these IGMP transactions. Protocol Independent Multicast (PIM) is then used between the local (R1) and remote multicast routers (R2), to direct multicast traffic from the multicast server to many multicast clients. IGMP operates above the network layer, just the same as other network management protocols like ICMP.

5

Testing and Results

OMNeT++ provides built-in support for recording simulation results, via output vectors and output scalars. Output vectors are time series data, recorded from simple modules or channels. Output vectors is used to record end-to-end delays or round trip times of packets, queue lengths, queueing times, module state, link utilization, packet drops, etc. anything that is useful to get a full picture of what happened in the model during the simulation run.

Output scalars are summary results, computed during the simulation and written out when the simulation completes. Statistics are declared in the NED files with the @statistic property, and modules emit values using the signal mechanism. The simulation framework records data by adding special result file writer listeners to the signals. By being able to choose what listeners to add, the user can control what to record in the result files and what computations to apply before recording.

Simulation Model

The simulation evaluations were carried out with the use of OMNeT++ (in version 4.0) simulation framework extended with the INET-HNRL package. The OMNeT++ is the modular, component-based simulator, with an Eclipse-based IDE and a graphical environment, mainly designed for simulation of communication networks, queuing networks and performance evaluation. The framework is very popular in research and for academic purposes. The INET – HNRL Framework is the communication networks simulation extension for the OMNeT++ simulation environment and contains models for several Internet protocols: UDP, TCP, SCTP, IP, IPv6, Ethernet, PPP, IEEE 802.11, MPLS, OSPF. INET-HNRL package provide the video server with trace files for different video codecs like H.264, H.265 and MPEG4. It also have module for VoIP and web traffic.

The INET-HNRL Framework builds upon OMNeT++, and uses the same concept: modules that communicate by message passing. Hosts, routers, switches and other network devices are represented by OMNeT++ compound modules. These compound modules are assembled from simple modules that represent protocols, applications, and other functional units. A network is again an OMNeT++ compound module that contains host, router and other modules. The external interfaces of modules are described in NED files.

Problems with Simulation model

INET-HNRL module had some compatibility issues with Omnet++ (ver. 4), so I had to make some changes to the INET module source file to make it compatible with the latest version of Omnet++. Omnet++ is still under development and in the latest version of Omnet++ VoIP streaming is still not included, as I needed the voice streaming for my simulation therefore I had to create a Voice streaming module for my project.

Modeling voice traffic as a CBR stream suffices, Traffic generator modules in the inet.applications, packages like UDPBasicApp, can generate both constant bit-rate and variable bit-rate traffic.

Simulation setup

Testing is divided into three parts. The simulated network is based on a subset sample of Telus network. Based on different situations the network is tested with Qos and without Qos. The network topology is presented in Fig 1. The link between R1 and R2 is the bottleneck of the network (changed as 80Mbps and 1Gbps). The evaluated Queue is the output queue of router R2. The queue is divided into three sub-queues for video, voice and web traffic. By adjusting the queue size a simple priority mechanism can be made. The number of clients is 30 and is same in all cases and data rate for different type of traffic is 3.3 Mbps for video, 1.2 Mbps for voice and 1 kbps for Data traffic.

Case 1: The simulation is performed without Qos and no queue management is deployed. It is further divided into two parts with congestion and without congestion. The link between R1 and R2 is changed as 80Mbps and 1Gbps.

Case 2: The simulation is performed with Qos and queue management is deployed. In this case we tried to increase the chances of video streaming over voice and data traffic, by giving more buffer to video as compared to voice and data. It is also divided into two parts with congestion and without congestion.

Case 3: The simulation is performed with Qos and H.264 is compared with H.265.

Result Analysis

The result analysis tool of OMNeT++ and its task is to help the user process and visualize simulation results saved into vector and scalar files. Result analysis tool is designed so that the user can work equally well on the output of a single simulation run.

The first page displays the result files that serve as input for the analysis. The upper half specifies what files to select, by explicit filenames or by wildcards. The lower half shows what files actually matched the input specification and what runs they contain.

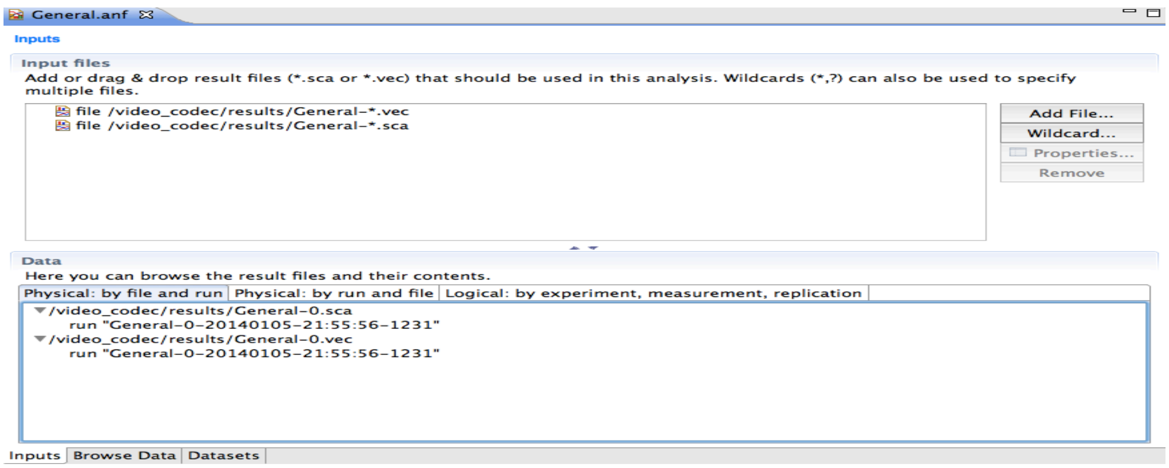


Figure 4: Input files for data analysis

The second page displays results (vectors, scalars, and histograms) from all files in tables and lets the user browse them. Results can be sorted and filtered.

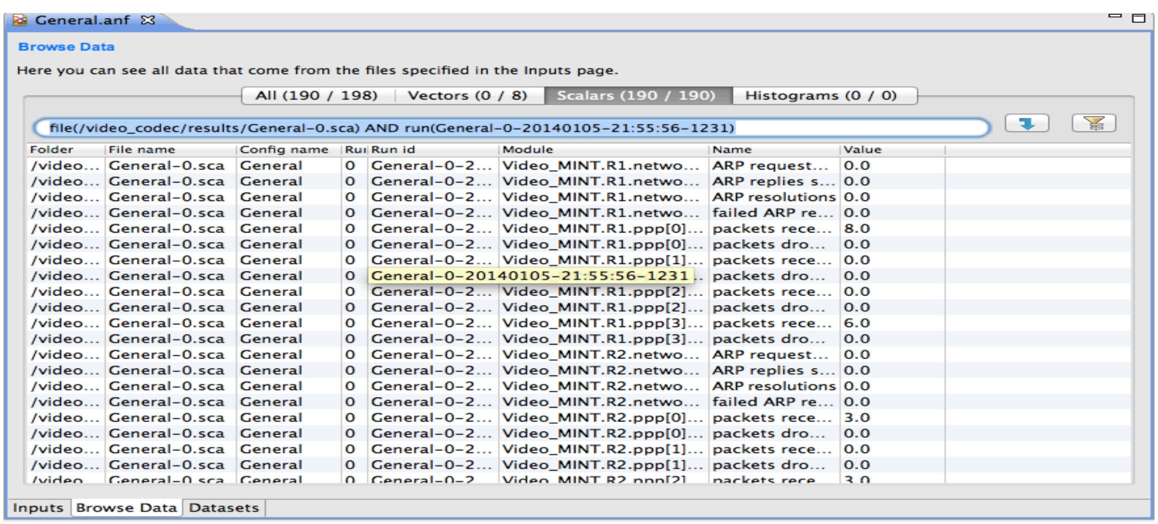


Figure 5: Browsing vector and scalar data

Sequence Chart

The OMNeT++ simulation kernel is capable of logging various events during simulation: scheduling and canceling self-messages, sending messages, display changes, module and connection creation and deletion, user log messages, etc. The result is an event log files which contains detailed information of the simulation.

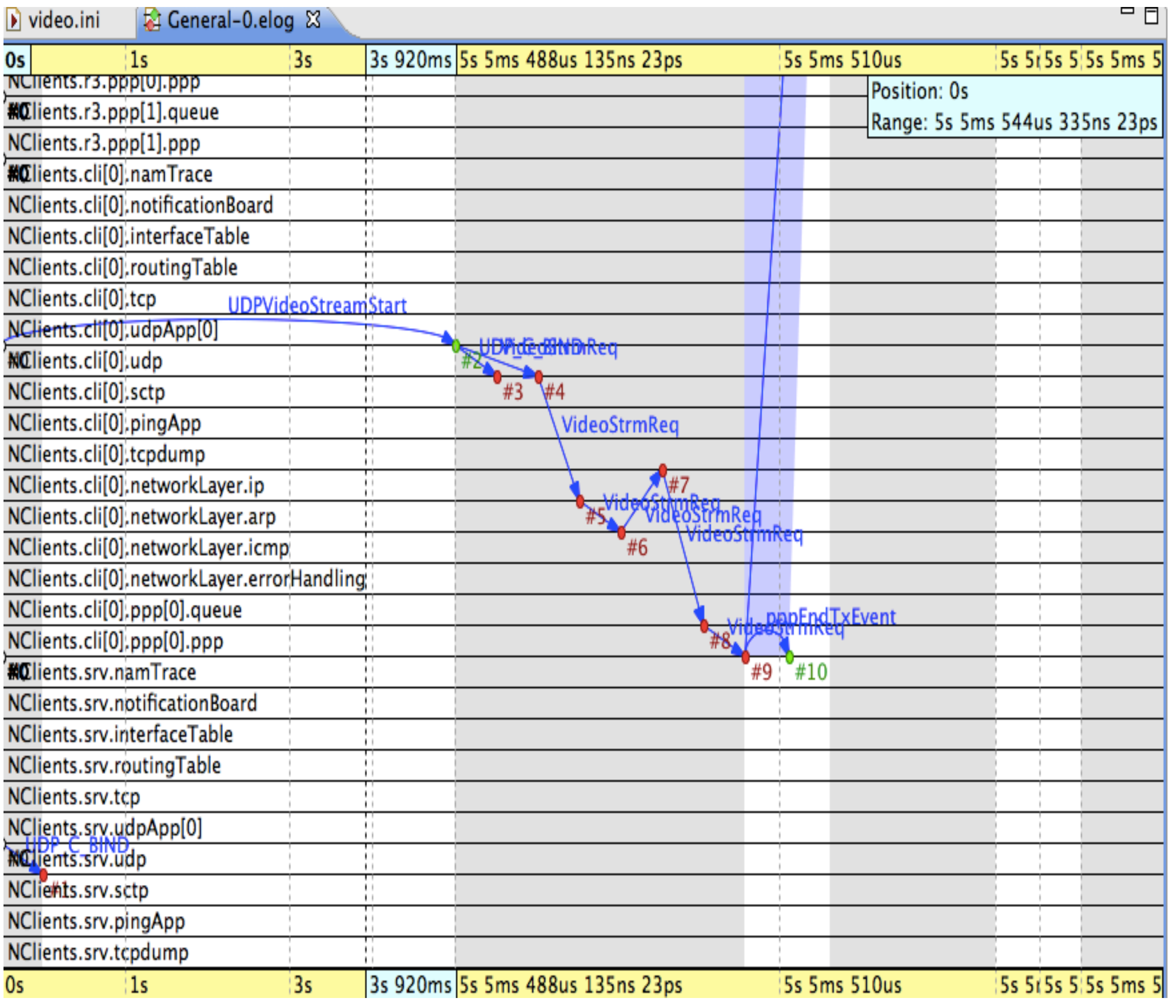
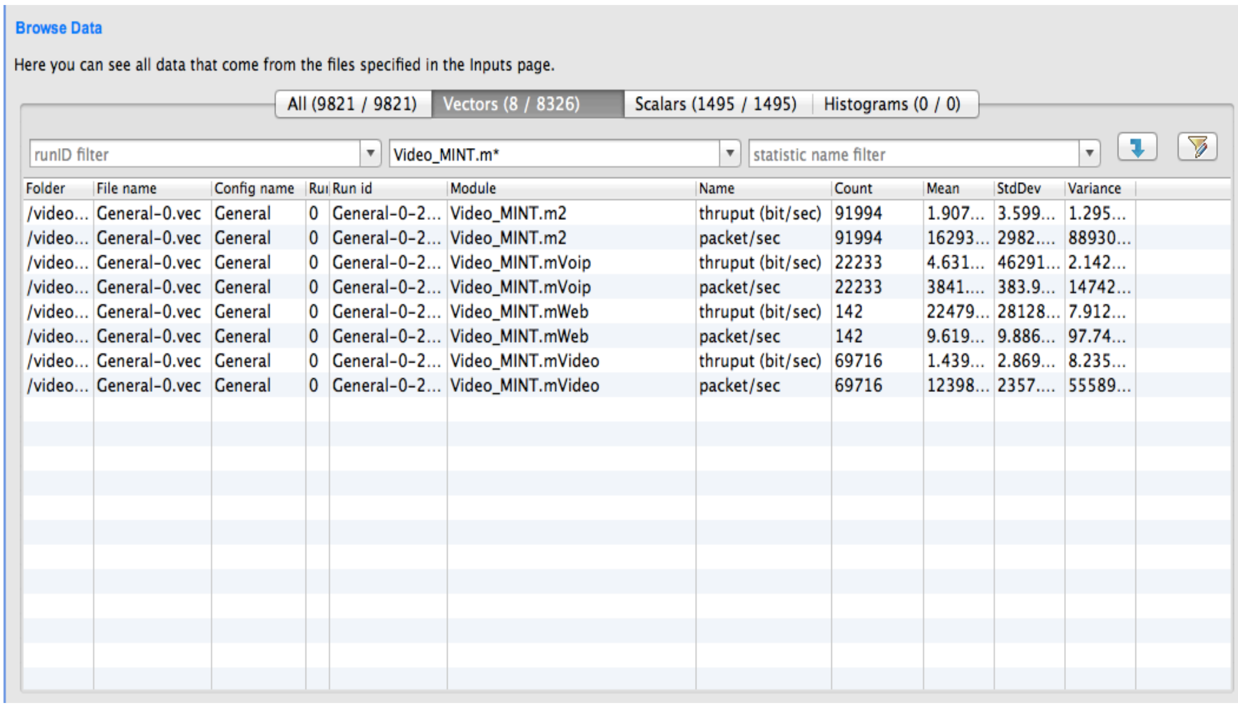


Figure 6: sequence chart showing udp stream

Evaluation of results

Now we are going to check the file general.anf, which will be very useful to graph the data taken across the simulation. We can create graphs with this information in Omnet++, which help us to get a better visualization. For example, if we are interested in showing the evolution of throughput in the simulation, we only have to select the vector throughput from browse data.



Browse Data

Here you can see all data that come from the files specified in the Inputs page.

All (9821 / 9821) | Vectors (8 / 8326) | Scalars (1495 / 1495) | Histograms (0 / 0)

runID filter | Video_MINT.m* | statistic name filter

Folder	File name	Config name	Ru	Run id	Module	Name	Count	Mean	StdDev	Variance
/video...	General-0.vec	General	0	General-0-2...	Video_MINT.m2	thruput (bit/sec)	91994	1.907...	3.599...	1.295...
/video...	General-0.vec	General	0	General-0-2...	Video_MINT.m2	packet/sec	91994	16293...	2982....	88930...
/video...	General-0.vec	General	0	General-0-2...	Video_MINT.mVoip	thruput (bit/sec)	22233	4.631...	46291...	2.142...
/video...	General-0.vec	General	0	General-0-2...	Video_MINT.mVoip	packet/sec	22233	3841....	383.9...	14742...
/video...	General-0.vec	General	0	General-0-2...	Video_MINT.mWeb	thruput (bit/sec)	142	22479...	28128...	7.912...
/video...	General-0.vec	General	0	General-0-2...	Video_MINT.mWeb	packet/sec	142	9.619...	9.886...	97.74...
/video...	General-0.vec	General	0	General-0-2...	Video_MINT.mVideo	thruput (bit/sec)	69716	1.439...	2.869...	8.235...
/video...	General-0.vec	General	0	General-0-2...	Video_MINT.mVideo	packet/sec	69716	12398...	2357....	55589...

Figure 7: Selecting data

1. **Without Qos:** In case 1 of simulation setup as suggested above, the simulation results are shown in this section.

   1.1 Without congestion: The link between R1 and R2 is congestion free and throughput for video, voice and data is measured using a throughout meter.
   The following graphs shows throughput for video, voice and data.

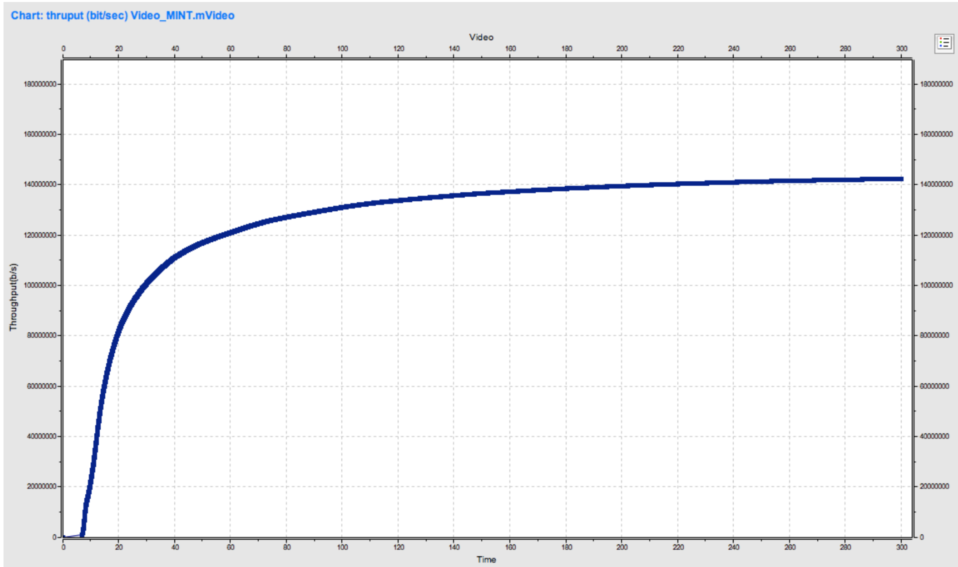


Figure 8: Throughput graph Video

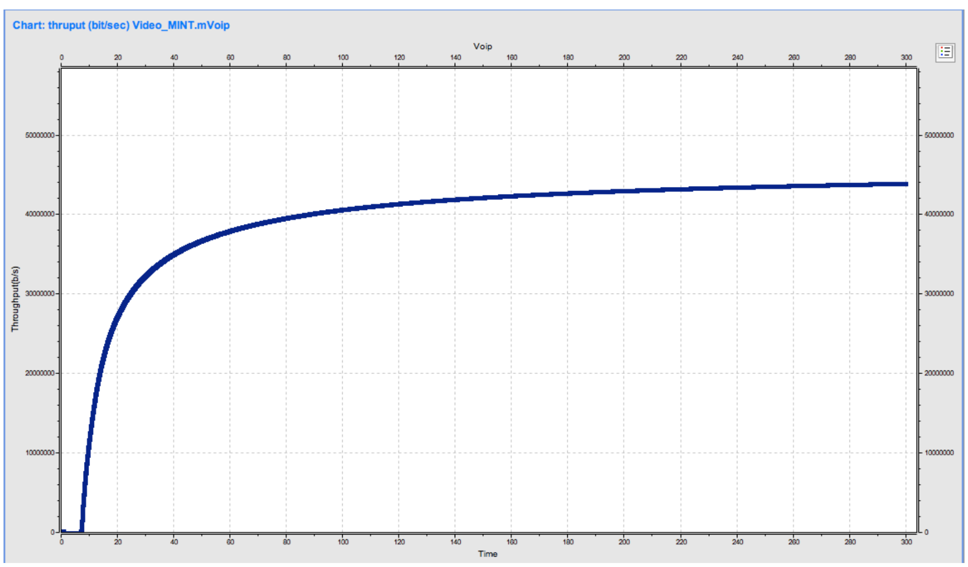


Figure 9: Throughput Graph Voice

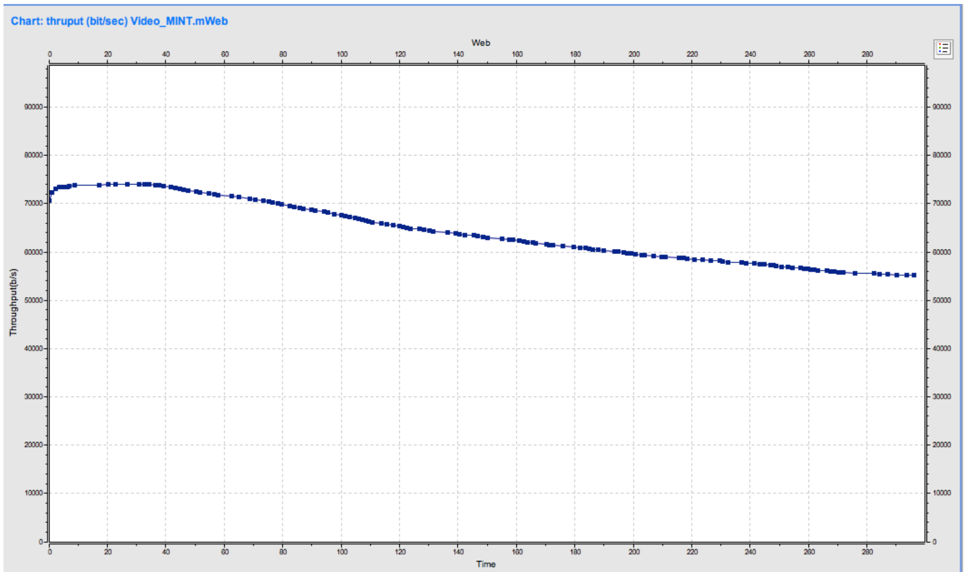


Figure 10: Throughput Graph Data

The above graphs show the throughput for different form of traffic. For video the average throughput measured is around 134.65 Mbps, for voice this value is 44.67 Mbps and for data it's 5.66 Mbps. End to end delay for video and voice is around 120ms and is measured using a delay meter. So without congestion and without Qos, these results show best effort service and as there is enough bandwidth available to traffic there is no packet drops.

1.2 With Congestion: the link between R1 and R2 is congested and as there is no Qos deployed, the router will try to give a best effort service to the traffic.
The following graphs shows throughput for video, voice and data.

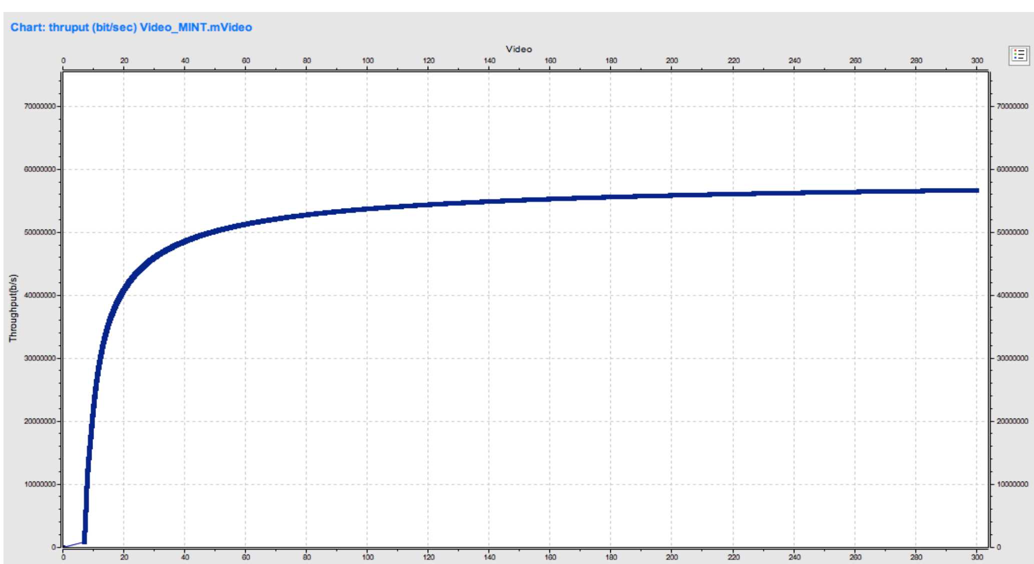


Figure 11: Graph for Video

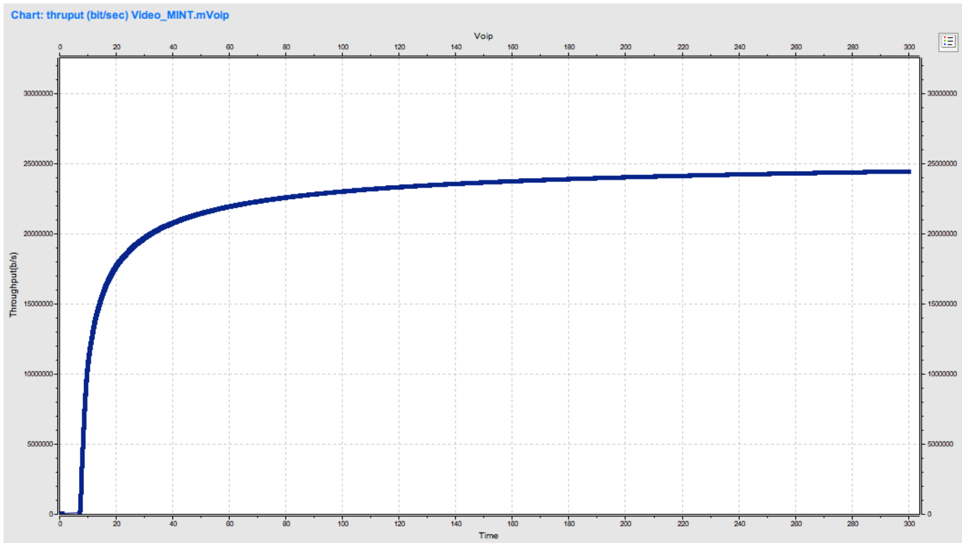


Figure 12: Graph for Voice

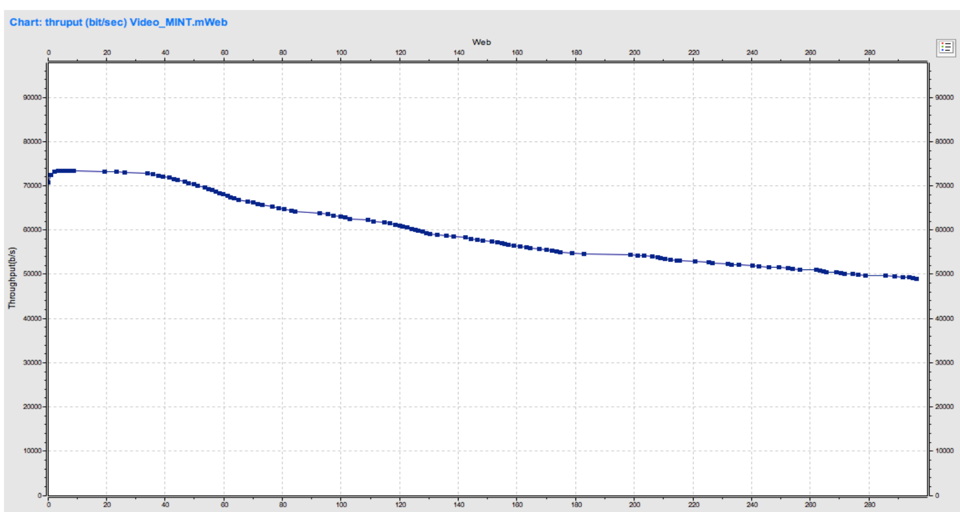


Figure 13: Graph for Data

The above graphs show the throughput for different form of traffic. For video the average throughput measured is around 50.18 Mbps, for voice this value is 25.63 Mbps and for data it's 4.12 Mbps. So with congestion and no Qos, these results show best effort service and as there is not enough bandwidth available to traffic there are packet drops. When there is congestion the routers are randomly dropping packets and try to give a best effort service. To give priority to Video traffic over voice and data we have deploy a Qos mechanism so that routers do not drop Video packets and Qos is assured for video traffic over voice and data.

2. **With Qos:** In case 2 of simulation setup as suggested above, the simulation results are shown in this section.

2.1 Without congestion: The link between R1 and R2 is congestion free and Qos is deployed. Throughput for video, voice and data is measured using a throughout meter. The graphs are similar to Fig 8, Fig 9, Fig 10

2.2 With Congestion: The link between R1 and R2 is congested and as there is Qos deployed, the router priorities the traffic according to queue management. The queue is divided into three sub-queues and each queue has a different queue frame capacity. Video has maximum capacity around 100 frames, voice has 25 frames and data 4 frames. The network capacity is 80 Mbps, the link is under congestion so Qos is applied. Queue is sub divided into three sub Queues for video, voice and data.

The size of queue depends on priority given to traffic, as video is getting more priority over voice and data. As our packet size is 1460 Bytes we can calculate packet rate on the network, 80Mbps/(1460*8)b which comes around 6849 Packet/s. so now we can calculate depending on the queue size the contribution of traffic from each Queue. Video queue size is 100, voice is 25 and data is 4, so if we calculate the percentage contribution from three queues. For video (100/129)=77.5%, for voice (25/129)=19.3% and for data (4/129)=3.1%.

Now we can calculate the data rate for traffic based on queue size. For video 6849*(77/100)=5273 Packets, if converted to Bit it comes around 5273*1460*8= 61.8 Mbps. And in simulation setup the average throughput measured for video is around 62.82 Mbps which is near to theoretical value measured.

The following graphs shows throughput for video, voice and data.

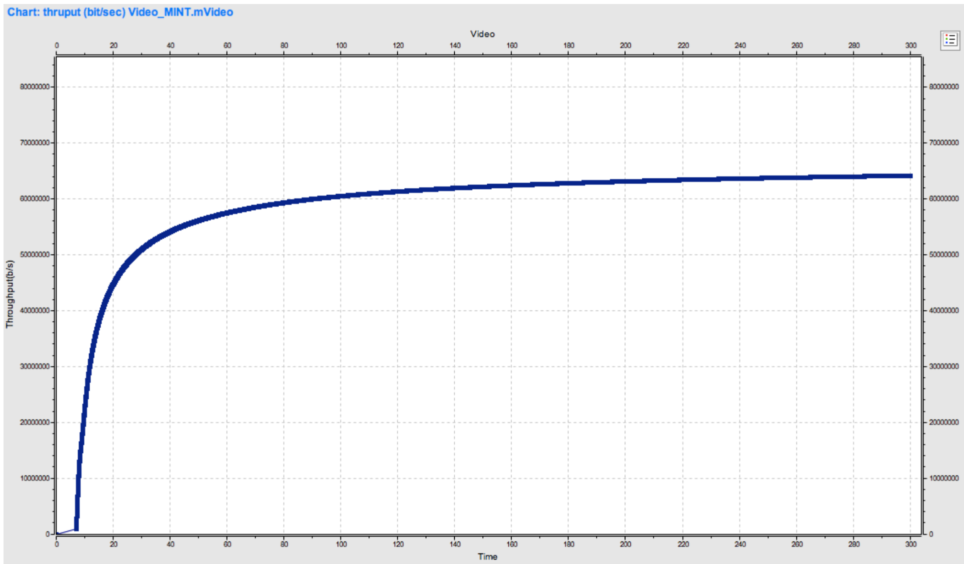


Figure 14: Graph for Video

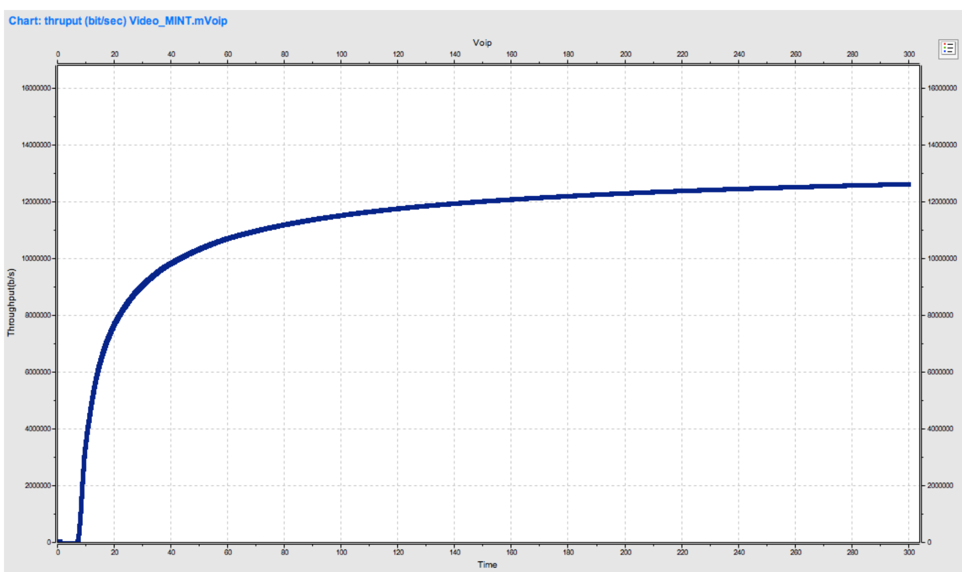


Figure 15: Graph for voice

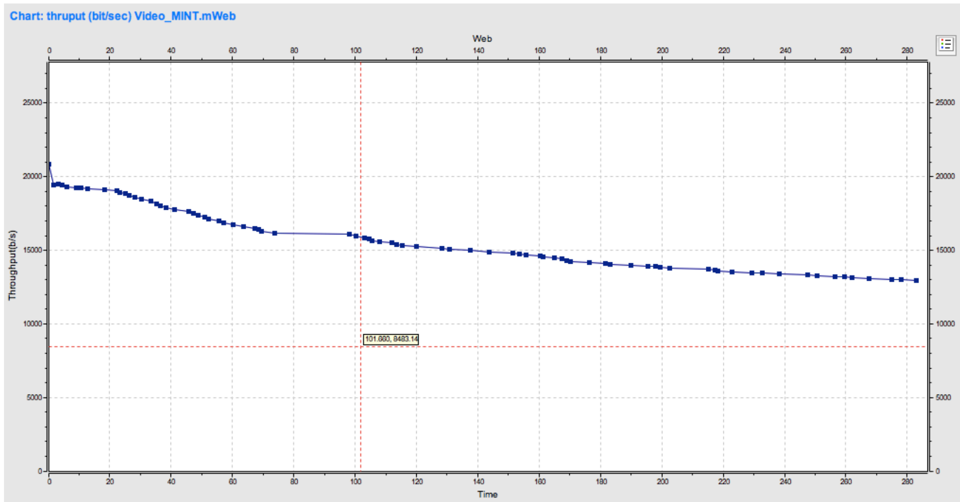


Figure 16: Graph for data

The above graphs show the throughput for different form of traffic. For video the average throughput measured is around 62.82 Mbps, for voice this value is 14.94 Mbps and for data it's 1.50 Kbps. So with congestion and Qos, these results show quality of service and as there is not enough bandwidth available to traffic there are packet drops for voice and data. When there is congestion the routers are dropping packets according to queuing management and giving priority to video over voice and data.

If we compare the throughput of fig 17 and 11, it's increasing from 50.18 Mbps to 64.82 Mbps, which represent video traffic. And between fig 18 and 12 it's decreasing from 22.62 Mbps to 16.94 Mbps. It shows that video is getting priority over video and data.

Bandwidth utilization table:

Bandwidth utilization	Without Qos		With Qos	
	No-congestion	Delay	Congestion	Delay
Video	135Mbps	120.02ms	62.82 Mbps	120.47ms
Voice	40 Mbps	120.04ms	14.94 Mbps	120.41ms
Data	5 Mbps		1.5 Kbps	

For end-to-end delay calculation delay meter is used, the delay meter calculates link delay and processing delay. The link delay is 40ms as there are three links between video and client server the link delay comes 120ms but the delay meter is showing around 120.47ms the 0.47ms increase is due to the processing delay added by switches and routers. Similarly for voice the delay is around 120.41ms.

3. **H.264 vs H.265:** The simulation setup is kept same and on video server the trace files are changed (H.264 and H2.65). Throughput is measured for both the codecs and graph are plotted. The graphs for H.265 codec reference Fig.14, Fig.15, Fig.16. The following graphs show throughput for H.264 codec.

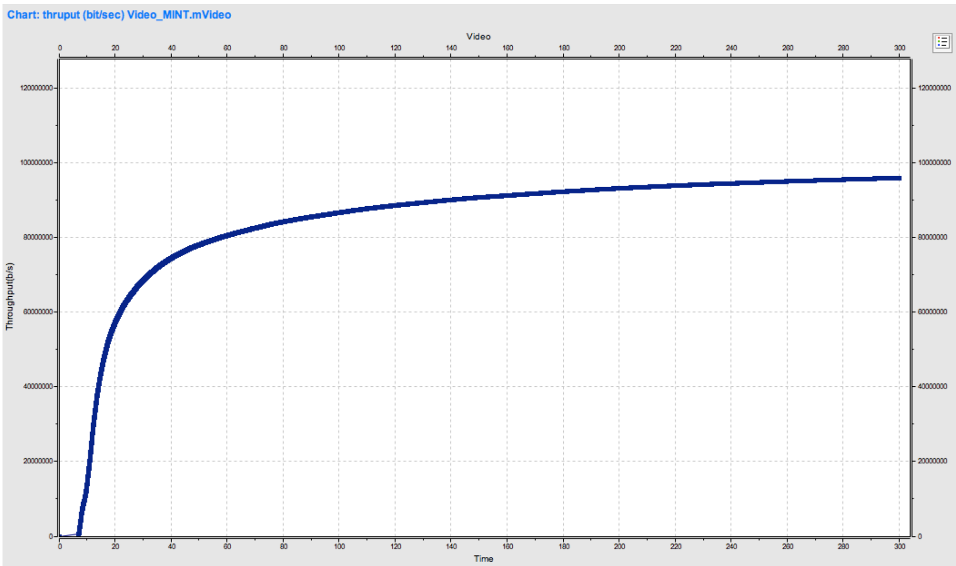


Figure 17: Graph for Video

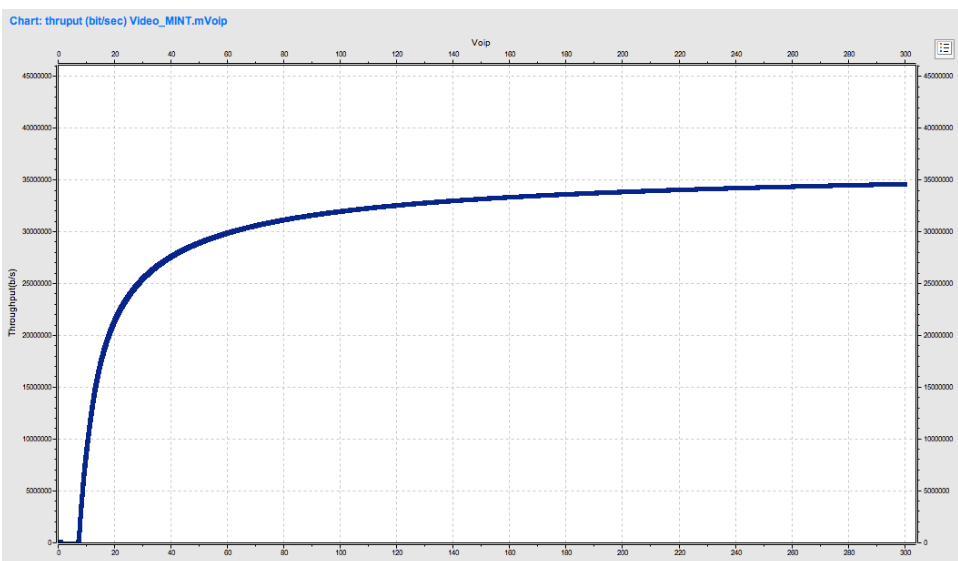


Figure 18: Graph for voice

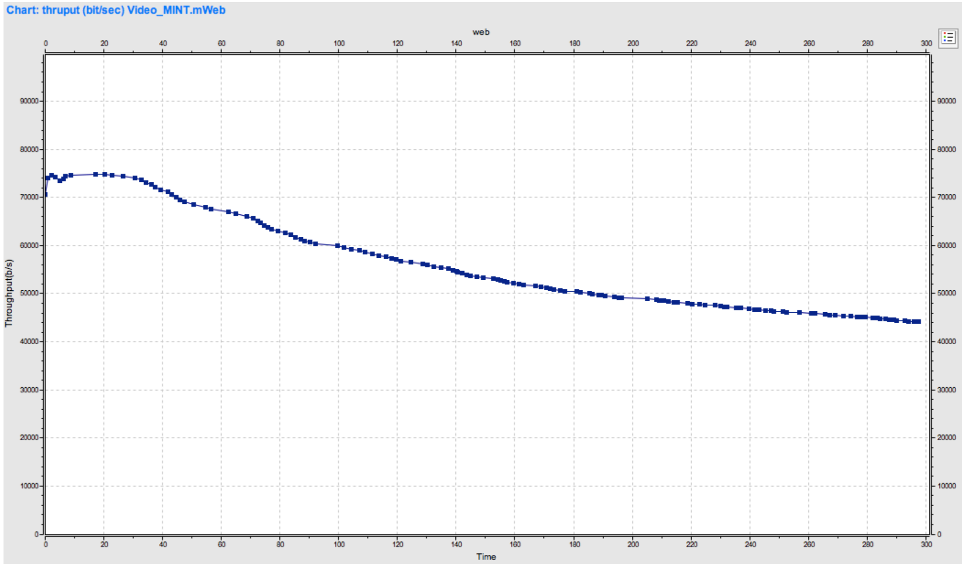


Figure 19: Graph for Data

Comparison between fig. 14 and fig. 20 show a difference in throughput measured. In Fig. 14 the average throughput for video traffic is 134.65 Mbps as compared to 97.65 Mbps in Fig. 20. The percentage difference is around 31.85%, where as the throughput for other traffic is almost same. The throughput for H.265 codec is around 31% higher than H.264.

6

Conclusion

This is my first time working in Omnet++, and I have to say that although in the beginning was very complicated because I didn't know to use the application, now I feel much self-confident and I know how powerful is working with it.

In this report we worked with simple examples of networks, we learnt to create a design, configure the .ini file, simulate and analyze results. The network was tested with Qos and without Qos, and findings are interesting. Without Qos it's a best effort service and router do not distinguish between packets and all packets are treated identical. In certain environment if a service provider wants to give video streaming priority over other form of traffic then Qos is deployed. Tests with Qos enabled helped in analyzing the traffic behavior and with a queue management video was given priority over voice and data. Results confirmed that Qos help in improving the quality of service.

H.265 is a new video codec and will replace the current H.264 in near future. Theoretically H.265 will cut the required bitrate by up to 50 percent, means throughput will increase by 50 percent. Test between H.264 and H265 shows that there is a significant increase in throughput. Both codecs were tested in identical conditions and results shows that there is around 31% increase in the throughput. Thus H.265 can support 4K – 8K videos, which is not possible with current codec used.

References

[1]. Patrick Seeling, Martin Reisslein, and Beshan Kulapala. *Network Performance Evaluation with Frame Size and Quality Traces of Single-Layer and Two-Layer Video*: A Tutorial. IEEE Communications Surveys and Tutorials

[2]. Geert Van der Auwera, Prasanth T. David, and Martin Reisslein. *Traffic and Quality Characterization of Single-Layer Video Streams Encoded with H.264/MPEG-4 Advanced Video Coding Standard and Scalable Video Coding Extension*. IEEE Transactions on Broadcasting, 54(3).

[3]. Patrick Seeling, Martin Reisslein "Video transport evaluation with H.264 Video Traces" PDF, IEEE Communications Surveys And Tutorials 14(4).

[4]. http://www.omnetpp.org/

[5]. Varga, András and Hornig, Rudolf (2008), "*An overview of the OMNeT++ simulation environment*", Simutools '08: Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops: 1—10

[6]. Kyeong Soo Kim, "*Integration of OMNeT++ hybrid TDM/WDM-PON models into INET Framework*," submitted to OMNeT++ Workshop 2011, Nov. 25, 2010

[7]. Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand "*Overview of the High Efficiency Video Coding (HEVC) Standard*"

[8]. *Advanced Video Coding for Generic Audio-Visual Services*, ITU-T Rec.H.264 and ISO/IEC 14496-10 (AVC), ITU-T and ISO/IEC JTC 1, May 2003 (and subsequent editions).