QoS Performance Analysis of Bit Rate Video Streaming in Next Generation Networks Using TCP, UDP and a TCP+UDP Hybrid

Xolani Dlamini
Postgraduate Student, Department of Computer Science, North West University Mahikeng Campus, South Africa

Francis L. Lugayizi
Postgraduate Student, Department of Computer Science, North West University Mahikeng Campus, South Africa

Bukohwo M. Esiefarienrhe
Associate Professor, Department of Computer Science, North West University Mahikeng Campus, South Africa

Abstract
The growth in users streaming videos on the Internet has led to increased demand for improved video quality and reception. In next generation networks (NGNs), such as 3G and 4G LTE, quality of service (QoS) implementation is one of the ways in which good video quality and good video reception can be achieved. QoS mainly involves following an industry-wide set of standard metrics and mechanisms to achieve high-quality network performance in respect of video streaming. Adopting routing and communication protocols is one way QoS is implemented in NGNs. This article describes QoS of bit rate video streaming, and QoS performance analysis of video streaming, in relation to the main network transport protocols, namely transmission control protocol (TCP) and user datagram protocol (UDP). A simulation test bed was set up using OPNET modeller 14.5. In this setup, a network topology was created and duplicated three times, in order to configure two simulation scenarios (each using the distinct protocols), and a third simulation scenario using both protocols in hybrid form. The findings in the simulations indicated that, when a network is configured with both TCP and UDP protocols in video streaming, there is a positive change in the degree of performance in terms of the QoS of video-streaming applications, unlike when the protocols are used independently.

Keywords
quality of service (QoS), bit rate video streaming, QoS routing, transmission control protocol (TCP), user datagram protocol (UDP)

DOI: https://doi.org/10.23962/10539/21759

Recommended citation
Dlamini, X., Lugayizi, F. L., & Esiefarienrhe, B. M. (2016). QoS performance analysis of bit rate video streaming in next generation networks using TCP, UDP and a TCP+UDP hybrid. *The African Journal of Information and Communication (AJIC), 18,* 135-154. https://doi.org/10.23962/10539/21759

1. Introduction

Next generation networks (NGNs) evolved from packet-switched networks that use Internet Protocol (IP) at the network layer (Knightson, Morita, & Towle, 2005). NGNs can either be wireless, such as for mobile phones, or wired, such as for desktops. A variety of applications make use of these networks, one of which is video streaming. Monitoring the performance of video streaming is considered one of the most challenging problems in NGNs (Adibi, Jain, Parekh, & Tofighbakhsh, 2010), requiring continual research and development. Quality of service (QoS) implementation and monitoring is one amongst many approaches that seek to evaluate the performance of video-streaming applications in NGNs. As more multimedia applications are developed and deployed onto these networks, it has become necessary to introduce more advanced mechanisms to monitor the performance of these applications, in order to achieve user satisfaction. Such mechanisms need to be able to enhance QoS in both legacy and concurrent NGNs, so to be able to meet user demand.

In this article, we describe QoS of bit rate video streaming, and then present a QoS performance analysis of video streaming in relation to the main network transport protocols, namely transmission control protocol (TCP) and user datagram protocol (UDP). The main research question we seek to answer is as follows: Is QoS evaluation a reasonable way to address performance challenges in bit rate video streaming in NGNs?

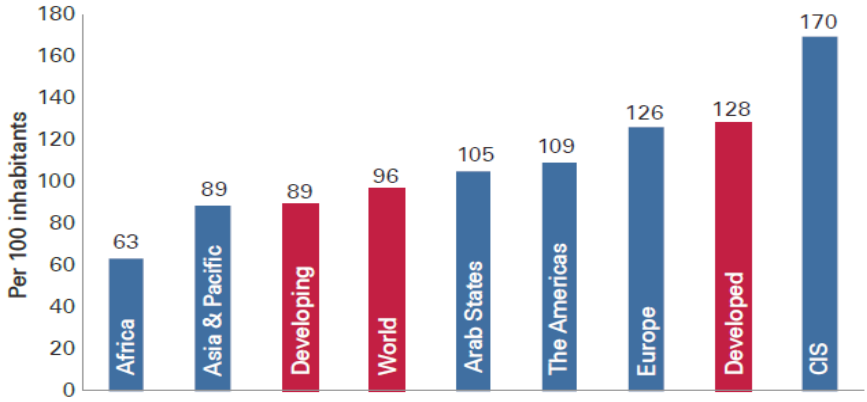
2. Background: QoS, TCP, UDP, bit rate video streaming

Mobile data traffic, namely for Internet access and video communication, increased three-fold every year in the 10 years between 2001 to 2010, and the demand for using mobile devices was seen to expand beyond 4.5 billion users (Cisco, 2010). By 2016, it was estimated that mobile data traffic had grown 4,000-fold over the previous 10 years, and that global mobile devices and connections had reached an estimated 7.9 billion in 2015 (Cisco, 2016). Video streaming, online games and IPTV are examples of the rapidly increasing need for real-time multimedia services. Furthermore, mobile ad hoc networks have grown abundantly in popularity, while the combination of mobile telecommunication networks with Internet continues to evolve in an innovative manner, via what are now termed next generation mobile networks (NGMNs) (Adibi et al., 2010). Evidence from the telecommunications industry (Tappayuthpijarn, Liebl, Stockhammer, & Steinbach, 2009) shows that NGMNs present a major advantage for operating networks, given the increase in digital traffic (Mok, Chan, & Chang, 2011). Some research shows that mobile videos will cause mobile data traffic to rise by more than 66%, expanding possibly 39 times, in the five years 2014 to 2019 (Freris, Hsu, Singh, & Zhu, 2013).

Figure 1 shows mobile growth around the world, which is challenging systems' capability and clients' connectivity. Increased mobile data traffic means increased

demand for QoS, for satisfactory user experience. QoS prioritises one type of traffic over another, helping to resolve data congestion in NGMNs.

Figure 1: International mobile growth



Source: Sanou (2013)

To meet an acceptable video quality, there is a need for a minimum bit rate. The bit rate measures how much data is being transmitted in a given peroiod of time, and the increased QoS demands can only be maintained if the required bit rates and delays are not too challenging (Ramanathan, 2005). Using multiple access networks to connect users to streaming servers is one of the techniques used to gain improved streaming quality. Two transport protocols dominate research discussions on streaming quality, namely TCP and UDP.

Parziale et al. (2006) present a summary of the invention of TCP, and explain how to create reliable, client-to-client transmission of data on a network. Previously TCP was considered as inadequate for video streaming. However, this has changed because it is now implemented with HTTP (Tappayuthpijarn, et al., 2009). TCP is the dominant protocol for high traffic volumes, and it promotes fairness between data transfers by evenly sharing the available bandwidth between users.

Unlike TCP, which is connection-oriented and involves "handshaking" between the network and devices, UDP is not connection-oriented and does not use handshaking dialogues. With UDP, there is no guaranteed delivery, meaning there is no repetition and ordering. Since UDP has no handshaking, it uses a normal model of transmission. For presenting functions that are different to source and destination of a datagram,

UDP only offers techniques for verification of data integrity. Accordingly, UDP has the following challenges: (i) it does not offer verification sequencing for datagrams; and (ii) it excludes connected datagram services. Therefore, it is important to note that source hosts that require consistent communication should use TCP (or a programme of similar reliabilty), which can provide own sequencing and acknowledge services (Zheng & Boyce, 2001), rather thatn useing UDP. This recommendation becomes a critical requirement in mobile video streaming applications within NGNs, especially when video streaming is of a real-time nature.

QoS measures

QoS is the process the network provider implements to deliver a satisfactory service, and with an assured level of service. To achieve this, QoS has to be measured. The main attributes that the metrics of QoS should always have are *timeliness*, *precision* and *accuracy* (Fiedler, Hossfeld, & Tran-Gia). *Timeliness* is the time taken to produce the result of the process. The number of results produced is measured by *precision*. The correctness of the results produced is a measure of *accuracy*.

In their work on QoS of computer networks and QoS measures, Mohapatra, Li, and Gui (2003) recommend that a few key QoS metrics are used to measure network end-to-end performance in relation to user requirements:

- ***Packet end-to-end delay:*** This refers to the elapsed time it takes for a packet to traverse from source, through the network, to its destination, and is measured in seconds. This is also referred to simply as end-to-end delay.
- ***Packet-delay variation (jitter):*** Whenever the end-to-end delay varies in a network, especially in video streaming, it is referred to as packet delay variation or jitter, measured in seconds.
- ***Bandwidth:*** Bandwidth refers to the highest rate of data transfer that a communication channel or link can sustain between a source and destination network. The difference between traffic sent and traffic received assists in determining the bandwidth available and in the long run, the point-to-point throughput of a channel. Traffic sent and received is measured in packets per second, while throughput is measured in bits per second.
- ***Packet loss (IP traffic dropped):*** This refers to a situation where a network loses data packets, especially where they fail to reach their destination network and is also referred to as IP traffic dropped. This is measured in packets per second.

QoS measures are generally implemented in video processing applications, due to the fact that video streaming is associated with constant delay requirements – and sometimes low bandwidth – and thus needs QoS interventions to provide quality.

QoS routing

QoS measures can be implemented through QoS routing, which is one of the driving forces behind video streaming applications. The main aim of QoS routing is to find

a path in a network that satisfies the given QoS limitations, such as energy, end-to-end delay and bandwidth availability. QoS routing is a scheme that takes into consideration the appropriate information about each link. Based on that information, it selects paths that satisfy the QoS requirements of a particular data flow (Asokan, 2010). Leela, Thanulekshmi, and Selvakumar (2011) state that the issue of QoS routing is crucial for dispersed applications, such as distributed games and Internet-enabled cellular phones. These dispersed applications place many different potential constraints on the aforementeiond QoS elements: *packet end-to-end delay, packet-delay variation* (*jitter)*, *bandwidth,* and *packet loss (IP traffic dropped)*. With QoS routing, the properties used to determine that one route is more appropriate than another are decided according to the QoS parameters. Routing in networks can either be *unicast* or *multicast*, as outlined below.

Unicast routing
The main feature of this routing algorithm is that it is used to connect only two nodes, namely a source and a destination, using a path that visits nodes in a predetermined way that corresponds to the location of routers (Dorigo & Stützle, 2003). Unicast is particularly efficient when video content delivery is among a group of limited users and using the point-to-point method. In unicast, a separate connection is created for each user, meaning resources are only used when the user of that given connection is active (Oyman, Foerster, Tcha, & Lee, 2010; Zhang & Wien, 2011). Typically, one of three main alternatives is used for the implementation of unicast routing: flooding, distance-vector routing, and link-state routing.

Multicast routing
In multicast routing, the algorithm simply states that one sender can send data to more than one recipient, and only one copy of the data is sent. Guo and Yang (2008) generated the idea of achieving the longest lifetime in mobile networks through two widely dispersed multicast routing algorithms. Multicasting is suitable for large numbers of users, as it can give good service even with large numbers of users streaming videos using mobile devices. (A routing mode is a single set of paths for sources and destinations in old networks, normally in the process of multicast routing.)

We now turn to a discussion of bit rate video streaming.

Bit rate video streaming
The volume of video traffic is expected to double annually in the coming years and, accordingly, to account for the dominant share of wireline and wireless Internet traffic (Cisco, 2010; 2016). In the past decade, various authors have reviewed the major topics in video streaming, such as scalable codecs, design of transport protocols, and adaptation techniques (see, for example, De Cicco, Mascolo, & Palmisano, 2011). Video streaming can be divided into two types: (i) live, real-time streaming,

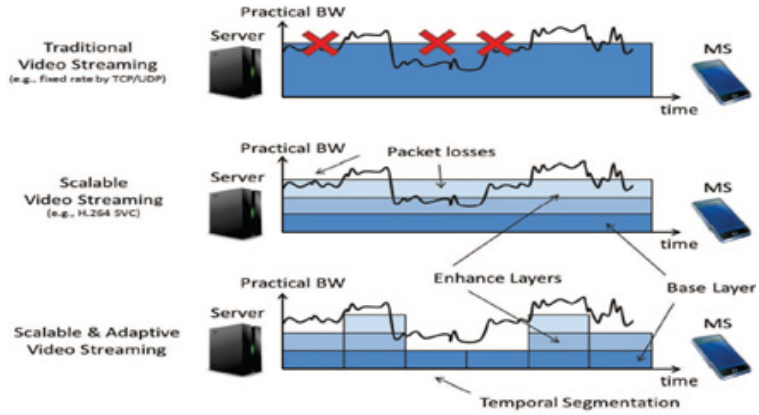
which is focused on encoding after capturing; and (ii) archived streaming, based on pre-encoding and storing for later viewing. Video conferencing applications, videophones, and interactive games are some of the examples of live, real-time video streaming. All these applications have strict delay requirements (Ji, 2009). Wireless networks are charactreised by a high bit error rate (BER) and frequent changes in channel quality (Fehér & Oláh, 2008; Tsai, Chilamkurti, Park, & Shieh, 2010), both of which are harmful to video communication, i.e., treaming quality can be harmed if the receiver tries to recreate the structure of the video from data characterised by errors. Thus the job of avoiding channel errors is central to QoS in video streaming over wireless networks.

Video codecs are designed to work at variable bit rate, i.e., a bit rate adjustable over long time scales by the video server. However, if the instantaneous wireless channel quality cannot support that bit rate (i.e., can deliver that bit rate only with a greater BER), video performance suffers dramatically (Aditya & Katti, 2011). Accordingly, variable bit rate encoding assigns more bits to complicated structures and fewer bits to less complicated structures, thus providing high video quality (Tabrizi, Peters, & Hefeeda, 2013). MPEG-4 is a compressing method in which the codec converts video traffic from low bit rate to high bit rate (see Memon, Hassan, & Memon, 2014). All multimedia traffic or video streaming has its own limitations that contradict the workflow of real-time protocols. For instance, multimedia traffic is a variable bit rate traffic origin, while the real-time networks are normally constant bit rate (CBR) channels, meaning that the amount of output of data per segment varies in multimedia traffic as opposed to real-time networks with a constant bit rate (Silvestre-Blanes, Almeida, Marau, & Pedreiras, 2011).

Some researchers have suggested use of context delivery networks (CDNs) as a streaming model. CDNs spread live or non-live video to users, i.e., before beginning a video streaming session (in a non-live setting), the source server can disperse a video to various assisted servers (Cisco, 2010; Nguyen, Nguyen, & Cheung, 2010). In order to view the video, the user then connects a few of these assisted servers in parallel.

Other bit rate approaches are scalable, and scalable and adaptive, video streaming. Figure 2 presents a comparison of traditional video streaming with scalable, and scalable and adaptive, methods (Chen, 2012). Alteration of link quality causes old video streams with fixed bit rates to fail to adapt to the changes, leading to packet loss and frequent termination of video streaming if the maintainable link bandwidth varies substantially a certain bit rate.

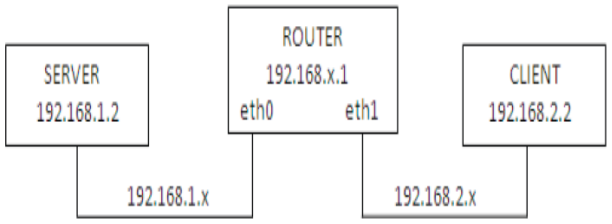
Figure 2: Traditional, scalable, and scalable and adaptive, video streaming



Source: Chen (2012)

Figure 3 shows a test setup used in Gürler and Bağci (2010). In this setup, the authors propose that content be streamed over managed local area networks (LANs), whereby the channel space available is randomly altered and impermanent introduction of the packets results in an additional 1% packet loss.

Figure 3: Network structures



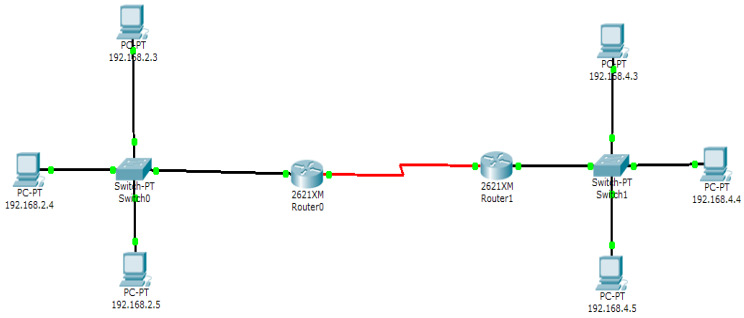
Source: Gürler and Bağci (2010)

By using video bit rate adaptation, the video quality can also be made fit for wireless networks. Khan, Sun, Jammeh and Ifeachor (2010) present a quality of experience-based model that can assist in adjusting the sending bit rate according to the supplied content. The quality of an output video sequence is evaluated by adopting

an evaluation model of video quality, which is defined by the number of decodable frames over the total number of frames originally sent from the video source (Lin, Ke, Shieh, & Chilamkurti, 2006).

As stated above, the protocols we used in our simulations in the test bed environment for QoS were TCP and UDP. Figure 4 shows a network that was designed on packet tracer, which uses the same transport protocols that were used in the test bed environment. As explained earlier, TCP is the more complex of the two protocols.

Figure 4: Sample network topology that uses transport protocols to transmit data



TCP is valued for its ability to open the shortest path first; to add reliability with retransmissions and ordering; and to offer fair bandwidth-sharing through congestion control. However, while TCP provides both reliability and ordering, and prevents congestion by controlling transmission rates, it is not optimised for video streaming (Lindeberg, Kristiansen, Plagemann, & Goebel, 2011). The TCP protocol is designed for wired networks and is not efficient for wireless networks. It reduces the transmission rate when there is a packet loss, which generates significant performance degradation in wireless networks because wireless channels generates high bit error rates. Source for TCP data maintains two "windows": a receive window for each destination, representing the available buffer capacity of each destination; and a "congestion window", representing the available capacity of the network. As the source transmits data, the size of each window is reduced by an amount equal to the size of the data sent (Shah & Patel, 2014).

UDP, the other dominant protocol in the computer networks environment, provides for less delay, but it increases packet loss because it has no network congestion avoidance mechanism.

TCP and UDP provide basic transport functions, while real-time transport protocol (RTP) and RTP control protocol (RTCP) run on top of TCP/UDP. UDP does not perform bandwidth adaptation or guarantee packet delivery, but it transmits the same bit rate as forwarded by the application (Hossfeld, Schatz, & Krieger, 2014). UDP is typically used by programmes that transmit small amounts of data at a time, or that have real-time requirements. In real-time situations, the low overhead and multicasting capabilities of UDP (for example, one datagram, many recipients) are better-suited than TCP.

This background discussion of QoS, QoS routing, and bit rate video streaming, has set the scene for presenting the experiment we conducted.

3. Test bed experimental setup and implementation

Our simulation consisted of testing three different routing protocols. The OPNET modeller 14.5 tool was used for the simulation, because it has the ability to generate accurate results for test scenarios via modelling traffic selection, projection and statistical data analysis, all of which require simulation. With its ability to enable designers to design either a small or large complex network, OPNET is a relatively powerful simulation software.

The network topology for the simulation was created in such a way that the two transport protocols (TCP and UDP) could be implemented. The sample applications used to generate video streaming traffic were a video conferencing application and a file transfer protocol (FTP) application. Since the focus of the experiment was on bit rate video streaming, the video conferencing application was set with a high-resolution video, because users have come to expect high resolution when streaming online videos. We ran all the simulations on Windows 7.0 Professional platform on a desktop with 3.40 GHz, 4.00GB RAM and a 32-bit operating system. Table 1 summarises the three scenarios created in the simulation test bed.

Table 1: Network simulation scenarios

	Scenario 1	**Scenario 2**	**Scenario 3**
Protocol	TCP	UDP	TCP+UDP
Application	Video	Video	Video
Application	FTP	FTP	FTP

The hybrid TCP+UDP arrangement was set up so that the advantages of each protocol could compensate for their respective disadvantages.

4. Results and discussion

This section presents the results of our simulations, and discussion of the QoS parameters – end-to-end delay; jitter; bandwidth, and IP traffic dropped – that were selected for the test bed experiment and used in the simulation to determine which of the two protocols (TCP and UDP) performed better in terms of video streaming, – or whether the third scenario using a hybrid of TCP and UDP performed better than either of the two protocols alone.

Packet end-to-end delay

Packet end-to-end delay is the time it takes for a packet to be transmitted across a network from source to destination. Statistics for packet end-to-end delay were selected for video conferencing before simulation and are displayed in Figure 5 below. End-to-end delay wwas calculated using this equation (1):

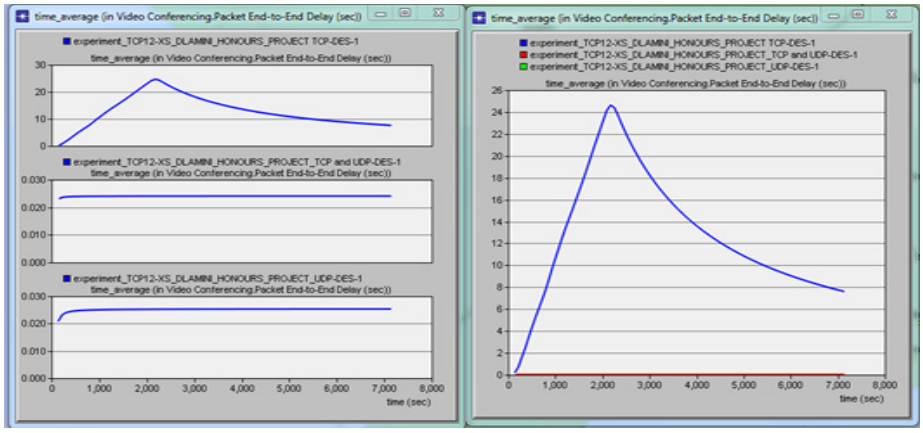
Dend_end = N (dtrans + dprop + dproc) (1)

(where dtrans is the transmission delay, dprop is the propagation delay, dproc is the processing delay, N is the number of links or routers, and Dend_end is the end-to-end delay)

In the simulation results shown in Figure 5 below, the TCP scenario is seen to have had the highest average time of packet end-to-end delay. The TCP three-way handshake characteristic clarifies the way TCP sends data, which may also cause delays if the destination takes time to acknowledge the source from which it received the sent information.

The UDP and TCP+UDP scenarios had equal amounts of packet end-to-end delay, and significantly less than in the TCP scenario. Unlike TCP, UDP does not guarantee packet delivery and does not establish a close connection, which causes its end-to-end delay readings to be low. In conclusion, the hybrid scenario was seen to be the best scenario for reducing packet end-to-end delay in streaming video on an NGN.

Figure 5: Packet end-to-end delay

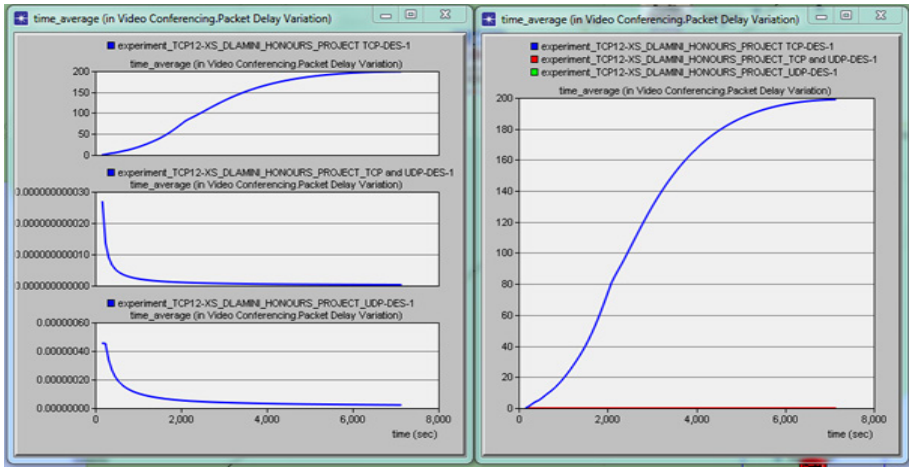


Note: *The red line of the TCP and UDP hybrid experiment is superimposed on the green line of the UDP experiment because the results have substantially similar values.*

Packet-delay variation (jitter)

Figure 6 below shows the different amounts of packet-delay variation that was noted during the simulation in each of the three scenarios. This variance in end-to-end delay for video packets and was measured from the time when a packet was created to the time when it was received.

Figure 6: Packet-delay variation (jitter)



Note: The red line of the TCP and UDP hybrid experiment is superimposed on the green line of the UDP experiment, because the results have substantially similar values.

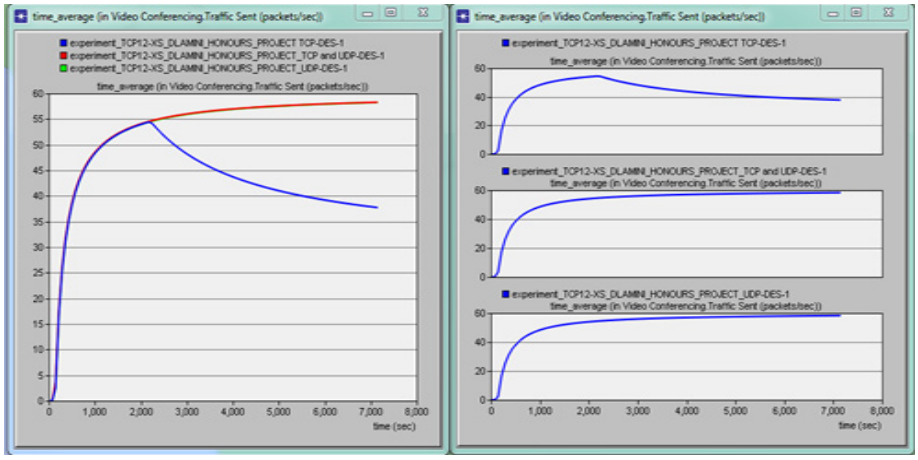
Packet-delay variation is the difference in end-to-end, one-way delay between selected packets in a flow, with any lost packets being ignored. Since this simulation concentrated on video streaming, packet delay was often present. Looking at the results, the TCP, UDP and TCP+UDP scenarios all improved packet delay. (Packet delay can be caused by having multiple hops. The experimental network topology had few hops, which could have caused the improvement in the three scenarios.)
The TCP scenario provided the best performance, generating less jitter than the UDP or TCP-UDP scenarios.

Bandwidth: Traffic sent

Traffic generated by each application was described in the "application definition" block of the OPNET modeller and since this work focused on video streaming, the video conferencing application was used. Traffic sent is the average number of packets per second submitted to the transport layer by, in this case, all video conferencing applications in the network.

Figure 7 below shows the results for the "traffic sent" statistics that were collected during the simulation. As the video conferencing application was the major source of traffic and video was accessed from the video-streaming servers, best-effort type-of-service and a frame size of 128 x 240 pixels were used.

Figure 7: Traffic sent



Note: The red line of the TCP and UDP hybrid experiment is superimposed on the green line of the UDP experiment, because the results have substantially similar values.

In Figure 7, the x-axis represents time in seconds and the y-axis represents the number of packets. The results were presented in a stacked (right-hand-side of figure) and

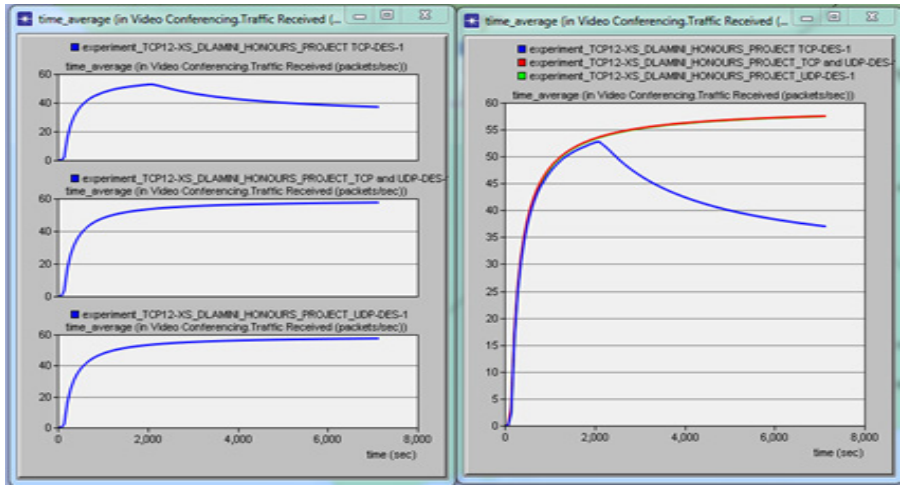
overlaid (left-hand-side of figure) form, so as to clearly display the differences across the three scenarios relative to the amount of traffic sent. The TCP simulation is represented in the topmost graph on the right of Figure 7, while the hybrid scenario is the middle graph and the UDP scenario is the bottom graph.

The downward slope of the curve in Figure 7 represents a drop in packets sent, which was a function of TCP, only noticeable from 2,300 seconds of simulation onwards. It is clear from these graphs that the weakest protocol, using the amount-of-packets-sent criterion, was TCP. Meawnhile, UDP and the hybrid TCP+UDP sent equal amounts of traffic and outperformed the TCP in terms of this criterion. As shown in Figure 7, the total number of packets transmitted by UDP and the hybrid TCP+UDP was 57.5 packets per second, while the total number of packets transmitted by TCP was 37.5 packets per second. For this criterion, it is thus better to use UDP alone than to use the hybrid approach, because no advantage is gained from the hybrid effort.

Bandwidth: Traffic received

Figure 8 below shows the amount of traffic received in the simulation experiments, where in each scenario the an equal amount of traffic was received and sent.

Figure 8: Traffic received



Note: The red line of the TCP and UDP hybrid experiment is superimposed on the green line of he UDP experiment, because the results have substantially similar values.

Traffic received in this simulation was the average number of packets per second forwarded to the video conferencing applications by the transport layers in the network. Figure 8 shows that there was very little if any difference in traffic received

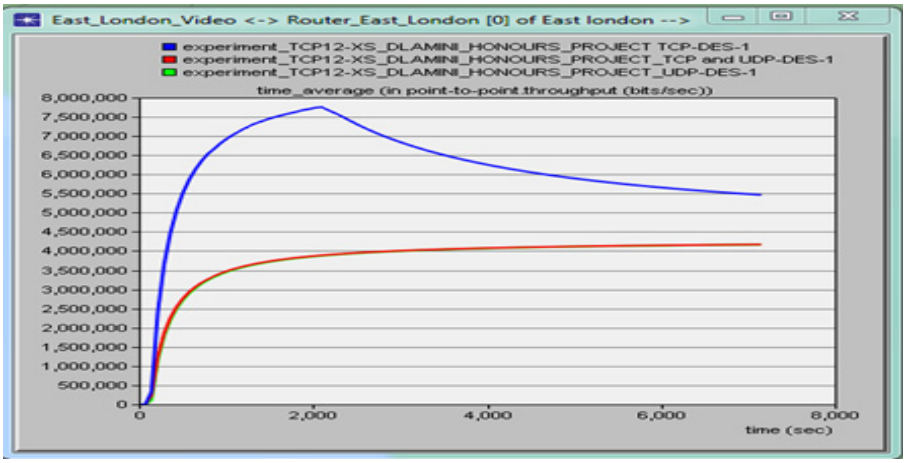
compared to traffic sent, in all three scenarios after 2,300 seconds has elapsed, though the UDP and hybrid scenarios received more traffic than the TCP scenario. The numbers of packets are presented on the y-axis against time on the x-axis.

In this study, our expectation was that the highest amount of traffic would be received when the TCP+UDP protocol hybrid was present. The average highest traffic amount in all scenarios was 57.5 packets per second, when the simulation was run for 7,200 seconds. These results show the rapid increase in the rate of traffic received, which necessitated an increase in the simulation time to two hours for better and more accurate results. At the end of the two-hour experiment, it was concluded that the traffic received in the UDP and hybrid scenarios showed better performance than in the TCP scenario. It was difficult to distinguish between performances in the UDP and hybrid scenarios, and thus we favour the UDP scenario because it did not require the additional effort for hybridising.

Bandwidth: Point-to-point throughput

Point-to-point throughput shows the time in relation to the average number of packets successfully received. Due to the fact that the work focused on bit rate, throughput was measured in bits per second. As can be seen in Figure 9, the results collected for all three scenarios were based on running the simulation for 7,200 seconds.

Figure 9: Point-to-point throughput



Note: The red line of the TCP and UDP hybrid experiment is superimposed on the green line of theUDP experiment, because the results have substantially similar values.

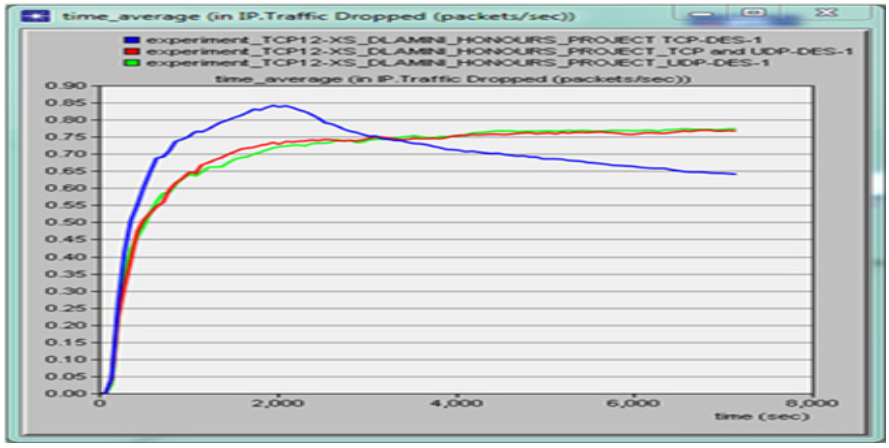
Point-to-point throughput is the average number of bits successfully received or transmitted by the receiver or transmitter channel per unit of time, in bits per second, which can also be referred to as the average rate of successful streamed videos from the servers to the clients in our network topology. The best throughput was found in the TCP scenario, followed by the TCP+UDP hybrid scenario, very closely followed by the UDP scenario.

(During the simulation process, the hybrid scenario was expected to have the highest point-to-point throughput, since it was using the features of both protocols. But in the results for the first QoS criterion (see Figure 5 on packet end-to-end delay), the TCP scenario had the highest end-to-end delay, which also influenced the high amount of point-to-point throughput in the same scenario.) Our findings thus suggest that configuring TCP is the best choice for video conferencing applications.

Packet loss (IP traffic dropped)

The IP traffic dropped statistic was chosen before running the simulations, so as to be able to compare the amount of traffic dropped across the TCP, UDP and TCP+UDP scenarios. Figure 10 below shows the data collected, measured in packets per second.

Figure 10: Packet loss (IP traffic dropped) in packets per second



IP traffic dropped is the number of IP datagrams dropped by all nodes in the network, across all IP interfaces. Handling traffic that enters a network can be carried out by controlling busy traffic and making sure that designated traffic flows get the correct bandwidth. This can often mean cutting off the excess flows, or changing the

precedence of the packets that exceed the bandwidth.

Traffic drops cause TCP to resend the packets, and Figure 10 shows that in the TCP scenario there was less packet loss than in the other two scenarios over 7,200 seconds. TCP also reduces the congestion window when it experiences great loss of packets. In all three scenarios, very few packets were lost at the beginning of the simulation, and the highest amount of IP traffic dropped in the TCP scenario was 0.84 packets per second, while the highest amount of IP traffic dropped in the TCP+UDP scenario was 0.76 packets per second.

Table 2 summarises the results obtained across the three scenarios.

Table 2: Results from the three scenarios

Video Conferencing	Statistics collected	Scenario 1 (TCP)	Scenario 2 (UDP)	Scenario 3 (TCP+UDP)
	Packet end-to-end delay (seconds)	25	0.025	0.025
	Packet-delay variation (jitter) (seconds)	200	0.000000045	0.000000000029
	Bandwidth: Traffic sent (packets/sec)	54	57.5	57.5
	Bandwidth: Traffic received (packets/sec)	54	57.5	57.5
	Bandwidth: Point-to-point throughput (bits/sec)	7,750,000	3,750,000	3,750,000
	Packet loss (IP traffic dropped) (packet loss/sec)	0.84	0.77	0.76

5. Conclusion

In this study, the main goal was to evaluate QoS, using TCP and UDP bit rate video streaming in NGNs, with the focus on the performance of the two transport protocols used. We started by first studying the challenges faced by NGNs and the benefits of using TCP and UDP when streaming videos. Secondly, we developed a framework imitating a real world network topology, where the implementation of the two transport protocols was carried out. Using the OPNET 14.5 modeller tool, we created three scenarios: one using TCP, the second using UDP, and the third using a hybrid of the two protocols.

We can categorically state that the key issue related to streaming videos online is bandwidth, which affects the streaming throughput and also network congestion. TCP is known to detect packet loss and when it is detected, TCP decreases the

congestion window and unnecessarily takes bandwidth from the competing traffic. However, use of a TCP+UDP hybrid enables more effective QoS, since UDP does not have the window congestion control capability.

Sent and received traffic were identical in all three scenarios. In the UDP scenario results, packet end-to-end delay and packet-delay variation (jitter) were low, while the TCP scenario presented the highest rate of throughput in bits per second. We conclude that, if a researcher or network administrator is concerned with high throughput when streaming videos, then TCP performs better than eith UDP or a TCP+UDP hybrid in delivery of efficient and effective network QoS.

6. Future work

In the literature review, we found that the number of mobile users streaming videos online has greatly increased. Accordingly, we now intend to extend our simulation by implementing the protocols TCP and UDP on a network congested with numerous mobile devices, and then examine QoS metrics. Moreover, we are interested in implementing this kind of QoS evaluation for streaming of live videos from different sources.

Finally, more work needs to be done on implementing these transport protocols on WiMAX technologies, which provide the same bandwidth as other wireless broadband NGNs but over longer distances with less interference.

References

Adibi, S., Jain, R., Parekh, S., & Tofighbakhsh, M. (Eds.) (2010). *Quality of service architectures for wireless networks: Performance metrics and management*. Hershey: IGI Global. Available at http://www.igi-global.com/book/quality-service-architectures-wireless-networks/37257 and http://www.igi-global.com/pdf.aspx?tid=99866&ptid=37257&ctid=15&t=preface

Aditya, S., & Katti, S. (2011). FlexCast: Graceful wireless video streaming. In *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*, (pp. 277-288). New York: ACM. Retrieved from http://web.stanford.edu/~skatti/pubs/mobicom11-flexcast.pdf

Asokan, R. (2010). A review of quality of service (QoS) routing protocols for mobile ad hoc networks. In *International Conference on Wireless Communication and Sensor Computing,* (pp. 1-6). New York: IEEE. doi: 10.1109/ICWCSC.2010.5415903

Chen, M. (2012). AMVSC: A framework of adaptive mobile video streaming in the cloud. In *IEEE Global Communications Conference*, (pp. 2042-2047). New York: IEEE. doi: 10.1109/GLOCOM.2012.6503416

Cisco. (2010, February 9). *Cisco visual networking index: Global mobile data traffic forecast update 2009-2014*. Cisco White Paper. San Jose, CA. Retrieved from http://theruckusroom.typepad.com/files/cisco-rmobile-trends-report.pdf

Cisco. (2016, February 3). Executive summary: The mobile network in 2015. *Cisco visual networking index: Global mobile data traffic forecast update, 2015-2020*. Cisco White Paper. San Jose, CA. Retrieved from http://www.cisco.com/c/en/us/

solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html

De Cicco, L., Mascolo, S., & Palmisano, V. (2011). Feedback control for adaptive live video streaming. In *Proceedings of the Second Annual ACM Conference on Multimedia Systems*, (pp. 145-156). New York: ACM. doi: 10.1145/1943552.1943573

Dorigo, M. & Stützle, T. (2003). The ant colony optimization metaheuristic: Algorithms, applications, and advances. In Glover, F. & Kochenberger, G. (Eds.). *Handbook of metaheuristics*. New York: Springer US, pp. 250-285. doi: 10.1007/0-306-48056-5_9

Fehér, G., & Oláh, I. (2008). Enhancing wireless video streaming using lightweight approximate authentication. *Multimedia Systems, 14*, 167-177. doi: 10.1007/s00530-008-0122-4

Fiedler, M., Hossfeld, T., & Tran-Gia, P. (2010). A generic quantitative relationship between quality of experience and quality of service. *IEEE Network, 24*(2). doi: 10.1109/MNET.2010.5430142

Freris, N. M., Hsu, C-H., Singh, J. P. & Zhu, X. (2013). Distortion-aware scalable video streaming to multinetwork clients. *IEEE/ACM Transactions on Networking, 21*(2), 469-481. doi: 10.1109/TNET.2012.2203608

Guo, S., & Yang, O. (2008). Maximizing multicast communication lifetime in wireless mobile ad hoc networks. *IEEE Transactions on Vehicular Technology, 57*(4), 2414-2425. doi: 10.1109/TVT.2007.912607

Gürler, C. G. & Bağci, K. T. (2010). Adaptive stereoscopic 3D video streaming. In *17th IEEE International Conference on Image Processing*, (pp. 2409-2412). New York: IEEE. doi: 10.1109/ICIP.2010.5651035

Hossfeld, T., Schatz, R., & Krieger, U. R. (2014). QoE of YouTube video streaming for current Internet transport protocols. In Fischbach, K. & Krieger, U. (Eds.) *Measurement, modelling, and evaluation of computing systems and dependability and fault tolerance*. Geneva: Springer International, pp. 136-150. doi: 10.1007/978-3-319-05359-2_10

Ji, G. (2009). VBR video streaming over wireless networks. Master of Applied Science thesis. University of Toronto. Retrieved from https://tspace.library.utoronto.ca/bitstream/1807/18771/11/Ji_Guang_200911_MASc_thesis.pdf

Khan, A., Sun, L., Jammeh, E., & Ifeachor, E. (2010). Quality of experience-driven adaptation scheme for video applications over wireless networks. *IET Communications, 4*(11), 1337-1347. doi: 10.1049/iet-com.2009.0422

Knightson, K., Morita, N., & Towle, T. (2005). NGN architecture: Generic principles, functional architecture, and implementation, *IEEE Communications Magazine, 43*(10), 49-56. doi: 10.1109/MCOM.2005.1522124

Leela, R., Thanulekshmi, N., & Selvakumar, S. (2011). Multi-constraint QoS unicast routing using genetic algorithm (MURUGA). *Applied Soft Computing, 11*(2), 1753-1761. http://dx.doi.org/10.1016/j.asoc.2010.05.018

Lin, C-H., Ke, C-H., Shieh, C-K., & Chilamkurti, N. K. (2006). The packet loss effect on MPEG video transmission in wireless networks. In *20th International Conference on Advanced Information Networking and Applications*, (pp. 565-572). doi: 10.1109/AINA.2006.325

Lindeberg, M., Kristiansen, S., Plagemann, T., & Goebel, V. (2011). Challenges and techniques for video streaming over mobile ad hoc networks. *Multimedia Systems, 17*, 51-82. doi: 10.1007/s00530-010-0187-8

Memon, S., Hassan, S. R., & Memon, N. A. (2014). Evaluation of video streaming performance over peer-to-peer network. In *International Conference on Collaboration Technologies and Systems*, (pp. 413-420). doi: 10.1109/CTS.2014.6867597

Mohapatra, P., Li, J., & Gui, C. (2003). QoS in mobile ad hoc networks. *IEEE Wireless Communications, 10*(3), (pp. 44-53). doi: 10.1109/MWC.2003.1209595

Mok, R. K., Chan, E. W., & Chang, R. K. (2011). Measuring the quality of experience of HTTP video streaming. In *12th IFIP/IEEE International Symposium on Integrated Network Management*, (pp. 485 492). New York: IEEE. doi: 10.1109/INM.2011.5990550

Nguyen, K., Nguyen, T., & Cheung, S-C. (2010). Video streaming with network coding. *Journal of Signal Processing Systems, 59*, 319-333. doi: 10.1007/s11265-009-0342-7

Oyman, O., Foerster, J., Tcha, Y-j. & Lee, S-C. (2010). Toward enhanced mobile video services over WiMAX and LTE [WiMAX/LTE update]. *IEEE Communications Magazine, 48*(8), 68-76. doi: 10.1109/MCOM.2010.5534589

Parziale, L., Britt, D., Davis, C., Forrester, J., Liu, W., Matthews, C., & Rosselot, N. (2006). *TCP/IP tutorial and technical overview*. 8th Edition. Armonk: IBM Redbooks. Retrieved from https://www.redbooks.ibm.com/redbooks/pdfs/gg243376.pdf

Ramanathan, R. (2005). Challenges: A radically new architecture for next generation mobile ad hoc networks. In *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking*, (pp. 132-139). New York: ACM. doi: 10.1145/1080829.1080843

Sanou, B. (2013). *The world in 2013: ICT facts and figures*. Geneva: International Telecommunication Union. Retrieved from https://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2013-e.pdf

Shah, S. M., & Patel, A. (2014). Enhance ADOV routing protocol in TCP and CBR traffic. *International Journal of Engineering Development and Research, 2*(1), 172-176. Retrieved from http://www.academia.edu/6678094/Enhance_ADOV_Routing_Protocol_in_TCP_and_CBR_Traffic

Silvestre-Blanes, J., Almeida, L., Marau, R., & Pedreiras, P. (2011). Online QoS management for multimedia real-time transmission in industrial networks. *IEEE Transactions on Industrial Electronics, 58*(3), 1061-1071. doi: 10.1109/TIE.2010.2049711

Tabrizi, F. M., Peters, J., & Hefeeda, M. (2013). Dynamic control of receiver buffers in mobile video streaming systems. *IEEE Transactions on Mobile Computing*, *12*(5), 995-1008. doi: 10.1109/TMC.2012.56

Tappayuthpijarn, K., Liebl, G., Stockhammer, T., & Steinbach, E. (2009). Adaptive video streaming over a mobile network with TCP-friendly rate control. In *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*, (pp. 1325-1329). New York: ACM. doi: 10.1145/1582379.1582670

Tsai, M-F., Chilamkurti, N., Park, J-H., & Shieh, C-K. (2010). Multi-path transmission control scheme combining bandwidth aggregation and packet scheduling for real-time streaming in multi-path environment. *IET Communications, 4*(8), 937-945. doi: 10.1049/iet-com.2009.0661

Zhang, B., & Wien, M. (2011). Towards robust video streaming for unicast and multicast: Modeling and implementation. In *17th IEEE International Conference on Networks*, (pp. 30-35). New York: IEEE. doi: 10.1109/ICON.2011.6168502

Zheng, H. & Boyce, J. (2001). An improved UDP protocol for video transmission over Internet-to-wireless networks. *IEEE Transactions on Multimedia, 3*(3), 356-365. doi: 10.1109/6046.944478