

# End-to-end QoS Measurement over a DVB-RCS Satellite Network

M. Ali, L. Liang, Z. Sun, H. Cruickshank, P. Thompson<sup>1</sup>, T. Bouquentar, and N. Alagha<sup>2</sup>

<sup>1</sup> CCSR, University of Surrey, Guildford, Surrey, GU2 7XH, UK,  
m.ali, l.liang, z.sun, h.cruickshank, p.thompson@surrey.ac.uk,

<sup>2</sup> European Space Agency, Netherlands,  
tajani.bouquentar, nader.alagha@esa.int

**Abstract.** Satellites play an important role in the future network due to their wide area coverage and for providing connectivity in remote regions of the world. This paper presents the end-to-end quality of service (QoS) measurements taken employing a European Space Agency (ESA) testbed over DVB-RCS infrastructure, in collaboration with University of Surrey, UK.

The applications chosen for these experiments are file transfer (FTP), web browsing (HTTP) and video streaming. File transfer and web browsing require reliable transport mechanism as a corrupted bit will hinder the intact data delivery. Therefore, these applications use transmission control protocol (TCP) as the transport protocol. TCP involves a three way handshake, which introduces extra delay during data transfer. Video streaming is a real time application. It is time-sensitive and requires lesser reliability compared to FTP and Web services. Hence, it employs user datagram protocol (UDP) at the transport layer, which do not offer any guarantee of reliable data delivery but timely. The parameters that have been used to evaluate quality of service (QoS) are packet delivery time, file download time, round trip delay, packet sizes and packet loss.

The paper presented measurement results and comparative analysis of the QoS of the applications over the DVB-RCS testbed.

**Key words:** QoS, DVB-RCS, Satellite network

## 1 Introduction

Satellite systems are integrated with the global information infrastructure as one of the major technologies providing both television broadcast services and Internet access. These systems are supporting the same Internet applications, services, and protocols as the terrestrial networks. Standards for satellite systems are required to be optimized for Internet access. Open standards have proven their merit in improving the telecommunications commercial market. In urban areas, Internet access is available via terrestrial networks, whereas, infrastructure deprived rural areas rely on wireless connectivity for this facility [8]. It takes a

long time to develop terrestrial networks to these rural areas. So, satellites can offer Internet access to such remote regions.

Satellites are characterized by their long propagation delay; hence, quality of service (QoS) needs to be handled correctly in these networks. In this paper, we are presenting the end-to-end QoS measurements taken in digital video broadcast-return channel via satellite (DVB-RCS) environment. The applications to be tested are file transfer (FTP), web browsing (HTTP) and video streaming and these applications will be used for the underlying services when taking measurements. The different QoS parameters considered are packet delivery time, file download time, round trip delay, packet size and packet loss.

This paper is divided into the following sections: Section 2 is an overview of DVB-RCS standard; Section 3 is a description of the ESA testbed used to perform the experiments; Section 4 describes the different applications used to perform the tests; the performance metrics are briefly reviewed in Section 5; the results have been analyzed in Section 6; and finally, Section 7 concludes the paper.

## 2 DVB-RCS Standard

The DVB Project (digital video broadcasting, DVB) started the development of a system for digital television broadcasting via satellite (DVB-S) [3] in 1992 and finalized the specification in 1993 [9]. Since then, the DVB-S has become the preferred format for satellite broadcasting worldwide. Initially, to support Internet services for DVB-S, the return channel used terrestrial networks. The emergence of digital video broadcast-return channel via satellite (DVB-RCS) [4] transformed the one-way DVB-S system into an interactive system and eliminated cables in the return path for Internet. The second generation of the DVB-S standard (DVB-S2) was finalized in 2006 with a considerable improvement in power and bandwidth efficiency. The DVB-S2 standard also allowed for Adaptive Coding and Modulation for interactive services with a significant gain in the overall system capacity and reduction of link margin.

The DVB-RCS standard was developed in 1997 and finally accepted as a European Telecommunications Standards Institute (ETSI) specification in March 2000. In September 2001, the guideline document was published. Early trial products were tested in 2000-2002, and the systems started getting stable, mature and widely acceptable during 2002. Satellite standardization organizations, operators, and manufacturers have made great progress over the last years to achieve a viable solution for interactive services via satellite [8].

In the forward link, DVB-RCS is based on the DVB-S and DVB-S2 standards. It employs time-division multiplexing at a data rate up to tens of megabits per second. In order to provide Internet access and interactivity, it uses the existing moving pictures expert group-2 (MPEG-2) transport stream to encapsulate IP packets. This encapsulation is performed according to the DVB specification. The return link in DVB-RCS employs a multiple-frequency time-division

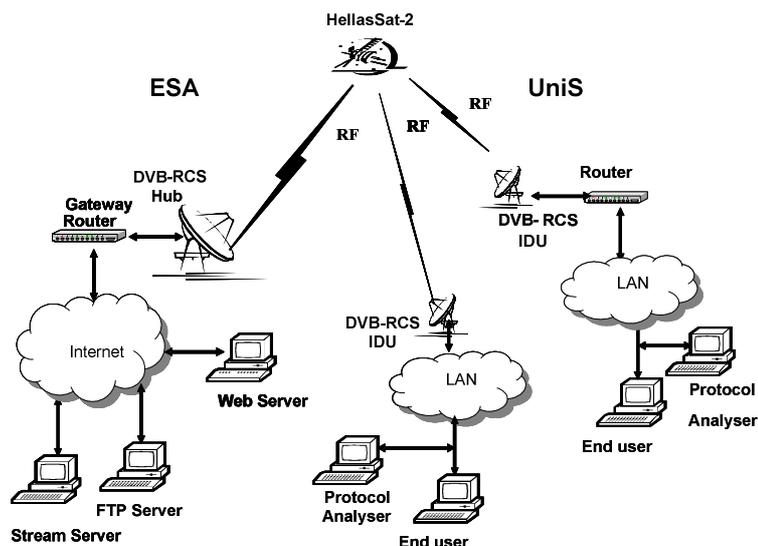


Fig. 1. ESA Testbed

multiple-access (MF-TDMA) scheme. It allows a group of terminals to communicate using a set of carrier frequencies, each of which is divided into time slots. The allocation mechanism is based on the statistical multiplexing principle, to utilize the satellite bandwidth optimally. This allocation is managed from the network control centre (NCC) at the gateway. The NCC is responsible for synchronization of the system, via the network clock reference (NCR). It transmits a number of specific system tables in order to update the terminals about issues related to quality of service (QoS), service level agreement (SLA), bandwidth allocation, etc.

### 3 The ESA DVB-RCS Testbed

A DVB-RCS testbed at the European Space Agency (ESA) is used for this end-to-end QoS measurement. As shown in Figure 1, the testbed consists of a gateway and two satellite terminals. The satellite capacity used in these experiments is provided by HellasSat-2 satellite. The available uplink bandwidth is 512 kbps and for downlink, it is 4 Mbps. Various applications and servers are installed on the local area networks (LAN) connected to the gateway and the terminals.

The testbed deploys performance enhancing proxies (PEP) to improve the performance of TCP based traffic. TCP deduces congestion in the network from delay and loss. Due to the long propagation delay in satellites, it takes a long time for TCP to increase its transmission rate and maintain a good throughput. PEPs split a single TCP connection into multiple connections. Splitting the connection allows to maneuver with different parts of the connection according to the condition of the satellite link [5].

## 4 Applications For Experimentation

Different applications that are tested on the testbed are described in the following subsections. The details of these applications are available in their corresponding standards, but an overview is necessary to understand the results. The softwares installed on the clients and servers are also described.

### 4.1 File Transfer

File Transfer Protocol (FTP) is a method of downloading files from and uploading files to another system using TCP over a network. It operates in a client/server paradigm. FTP uses two separate connections to send files. One TCP connection is used to send control messages, such as user identification, password, etc., and the other is used to send data. As FTP sends the control messages separately, so it is referred as an out-of-band protocol. There are a number of commands used in FTP for managing the sessions. The commands, from client to server, and replies, from server to client, are sent across the control connection in 7-bit ASCII format. Thus, these commands are human readable. Each command consists of four uppercase ASCII characters, some with optional arguments. Some of the more common commands are listed below:

- USER username: Used to send the user identification to the server
- PASS password: Used to send the user password to the server
- LIST: List all the files in the current directory in the server
- RETR filename: Used to retrieve (i.e., get) a file from the current directory of the server
- STOR filename: Used to store (i.e., put) a file in the current directory of the server

Each of these commands is followed by a reply from the server. The replies are three-digit numbers, with an optional message following the number. Some typical replies, along with their possible messages, are listed below:

- 331 Username OK, password required
- 230 User logged in, proceed
- 200 Command OK
- 125 Data connection already open; transfer starting
- 150 File status OK; about to open data connection
- 226 Transfer complete
- 221 Goodbye

A client utility (`ftp`) invokes the file transfer. In addition to the original `ftp` utility, there are many textual and graphical FTP client programs, including most browsers, which run under different operating systems. We used `ftp` as the client program for the end-to-end QoS measurements. There are also many FTP server programs. One of the available servers included in Linux is Very Secure FTP daemon (`vsftpd`) package.

## 4.2 Web Browsing

Web browsing is based on the HyperText Transfer Protocol (HTTP). It also works in a client/server model. The client and server programs exchange HTTP messages to transfer web content. The client program is the browser that opens the Web pages and offers other features like playing audio and video, uploading and downloading files, filling forms, etc. Popular Web browsers are Firefox, Netscape, lynx and Internet Explorer. On the server side, the Web server, houses the web content. HTTP uses TCP as the transport protocol. One of the world-wide deployed, Linux based Web server is Apache. In its latest versions, if support for IPv6 is available in the operating system, its IPv6 listening sockets open by default [6].

Like file transfer, HTTP also exchanges request and response messages. The typical methods and their purpose in a request message are as follows:

- GET: Request an object from the server
- POST: Filling form on a web page
- HEAD: Similar to GET used for debugging

The majority of the HTTP request messages use the GET method to retrieve objects from web servers during browsing. In the HTTP response messages, the replies are identified by the status code. Some of the commonly used status codes are listed below:

- 200 OK: Request succeeded and the information is returned in the response
- 301 Moved Permanently: Requested object has been moved permanently
- 400 Bad Request: The request could not be understood by the server
- 404 Not Found: The requested document does not exist on this server

## 4.3 Video Streaming

In this class of applications, clients request on-demand compressed video files that are stored on servers. A client typically begins playout of the video a few seconds after it begins receiving the file from the server. The client will be playing out video from one location in the file while it is receiving later parts of the file from the server. Both TCP and UDP are used in video streaming.

For the purpose of our test, VLC media player [1] is used. It is a lightweight media player that can play numerous video formats. It is supported by Windows, Linux and PocketPC/WinCE handhelds. It can also be used as a streaming server.

## 5 Performance metrics

The performance of different applications described, is measured, using the ESA DVB-RCS testbed. Wireshark [2], formerly known as Ethereal, a packet analyzer is used to capture packets at the client and server to measure and calculate different performance metrics. The end-to-end QoS parameters measured are described in the following subsections.

### 5.1 Delay

Delay is the transit time between the client and the server observed by the packets. In satellite networks, this time is dominated by the propagation delay. The processing time for the encapsulation of IP packet in MPEG stream also adds to the overall delay. In addition to this, other factors adding to the delay are router queues and number of hops in the path. A simple way to calculate the delay is to synchronize the client and the server and calculate the time difference. One way for synchronization is to use Network Time Protocol (NTP) server [7].

### 5.2 Jitter

Jitter is the variation in delay of consecutive packets. It is due to the variable time elapsed in the queues of the routers in the path. Packet time-stamps are used to measure jitter. The delay experienced in router queue is dependent on the scheduling policy of the router and the congestion in the network. In satellite networks, devices like modem, the modulator in the hub and the IP encapsulator also introduce jitter. Another similar parameter dependent on time-stamps is packet inter-arrival time.

### 5.3 Packet Size

Packet size affects different applications. Larger packets experience more packet loss and delay as compared to smaller packets. Smaller packets are vulnerable to higher jitter in contrast with larger ones.

### 5.4 Packet Loss

Packet loss occurs due to network congestion and buffer overflow at routers. In satellites, it is due to the wireless nature of the transmission link. In satellite networks, it could be due to weather conditions, improper line up, interference, etc. It can be reduced by retransmission or error recovery mechanisms. Some applications cannot tolerate packet loss beyond a certain threshold.

## 6 Results And Analysis

The experiments have been performed for different applications as mentioned earlier. These applications are analyzed in the following subsections.

### 6.1 File Transfer

For analysis of file transfer using file transfer protocol (FTP), two files of different sizes have been chosen to compare the results. One file is 500 kilobytes and the second is 50 megabytes. These files have been uploaded (via put command) from

**Table 1.** Round Trip Time for Request and Response pairs (milliseconds)

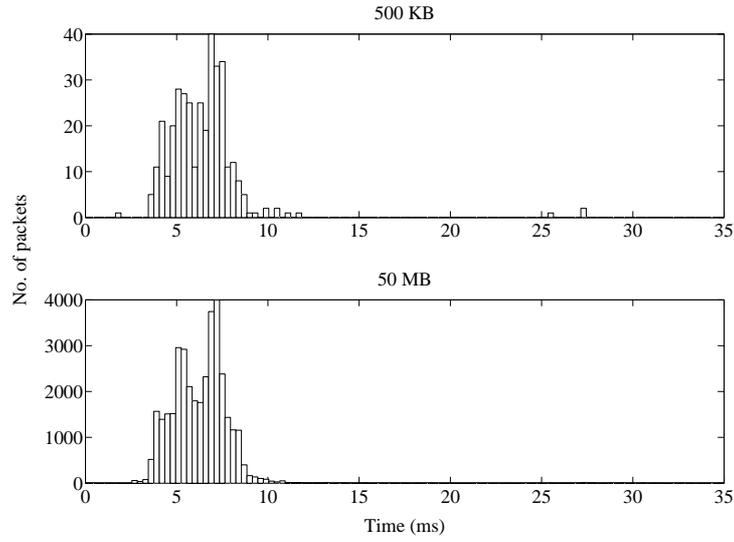
FTP Request	FTP Response	Upload (put)		Download (get)	
		500 KB	50 MB	500 KB	50 MB
USER	331 Password required	620.621	610.481	604.685	652.596
PASS	230 Access granted	748.752	701.774	724.554	944.772
PORT	200 PORT Successful	618.772	596.778	608.898	613.617
STOR/ RETR	150 Opening BINARY mode	680.763	629.675	680.800	687.860
QUIT	221 Goodbye	615.153	662.917	605.296	618.140
Total upload/download time (seconds)		18.470	2085.830	1.640	228.040
Transfer rate (kbps)		221.600	196.400	2511.760	1796.160

client (Surrey) to server (ESA) and downloaded (via get command) in the reverse direction.

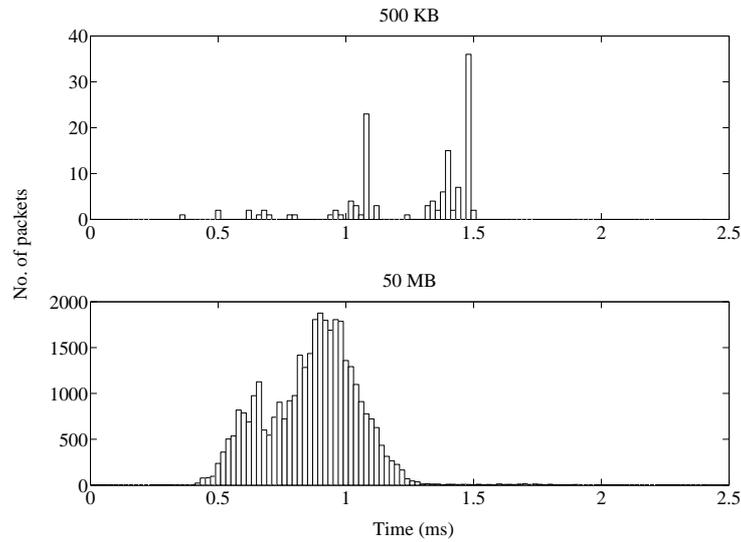
The round trip time (RTT) for the control commands for these operations is tabulated in Table 1. The RTT ranges from 550-700 milliseconds, which is the typical range for a geostationary satellite. The RTT values in the second row of Table 1 are higher as these values include the password typing time from the user. The upload time of the files is 10 times the download time because of the difference in the uplink and downlink data rates offered by the satellite network. The transfer rate in downloading is high than uploading due to the high bandwidth available on the downlink.

The role of performance enhancing proxies (PEP) deployed in the satellite network can be depicted from the histograms shown in Figure 2 and 3. These histograms have been generated from the FTP data being uploaded/downloaded and the acknowledgements being received from the PEP rather than from the recipient. As very few packets are exchanged while downloading the 500 KB file, hence, the density of packets is very low than during upload, resulting in a different shape of histogram. Similarly, less packet exchange can be depicted from the vertical scale of the figures during download of the 50 MB file, as compared with the upload.

The IP packet length of different packets during the upload and download is shown in Figure 4 and Figure 5, respectively. The packets on the right side show the packets depicting the FTP data and the packets on the left side are

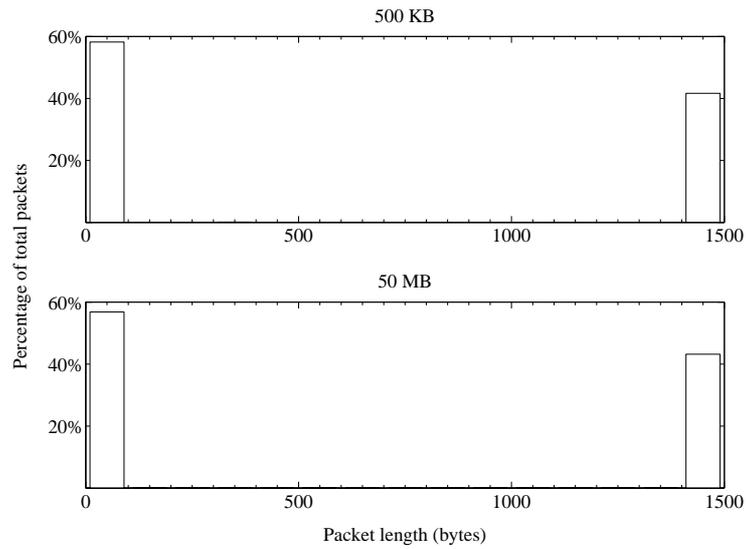


**Fig. 2.** RTT for ACK from PEP in upload of a 500 KB and a 50 MB file

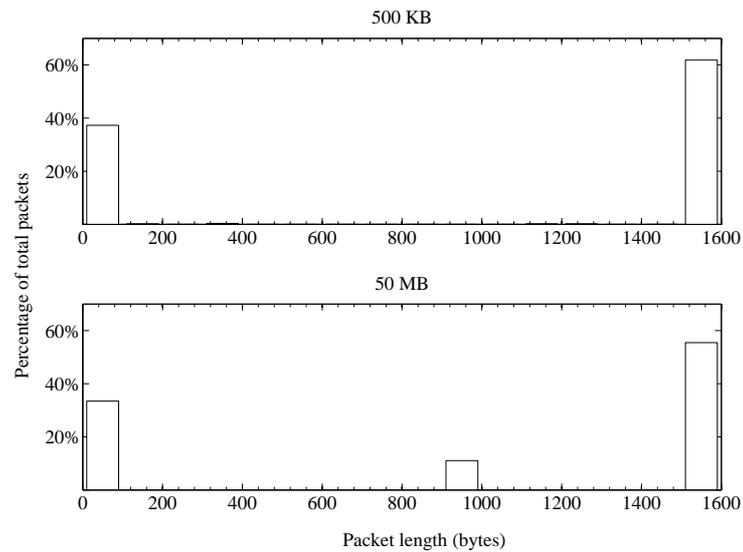


**Fig. 3.** RTT for ACK from PEP in download of a 500 KB and a 50 MB file

the TCP acknowledgements. Some packets in between are the last portion of the FTP data and control commands. It can be deduced from these results that during upload, the percentage of TCP ACKs is more than the FTP data. It is understood that this is due to the lower bandwidth available on the uplink resulting in more TCP retransmissions and duplicate ACKs. On the other hand,

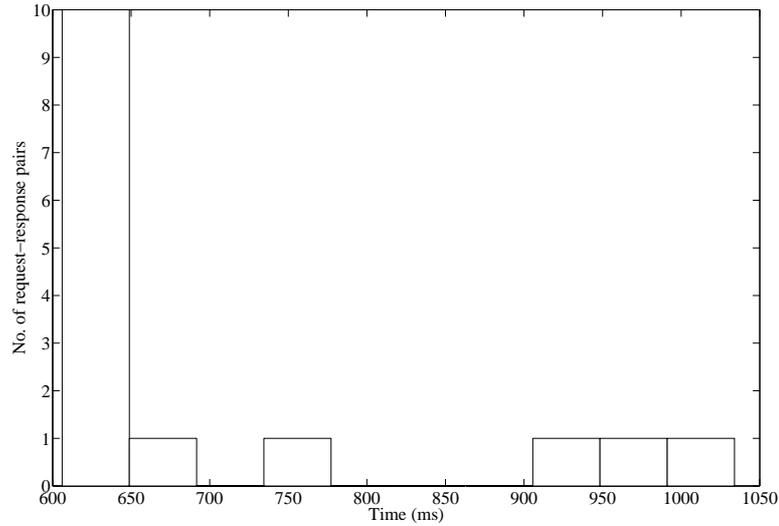


**Fig. 4.** IP packet length during upload of a 500 KB and a 50 MB file



**Fig. 5.** IP packet length during download of a 500 KB and a 50 MB file

during download, plenty of bandwidth is available, so more FTP data can be sent and less ACKs are required due to the cumulative acknowledgement phenomenon of TCP. In downloading the 50 MB file, mostly FTP data packets are 1500 bytes and quite a lot of them are 932 bytes, so there is a prominent bar near 1000 byte scale on the histogram in Figure 5.



**Fig. 6.** RTT for HTTP request-response pairs

During the upload and download of the 500 KB and 50 MB files, most of the frames have inter-arrival time in the microsecond and millisecond range. Some packets exhibit more inter-arrival time depicting the user password typing time, the time taken by server to open ports for data transfer during execution of PORT command, total file transfer time and user idleness to execute the QUIT command.

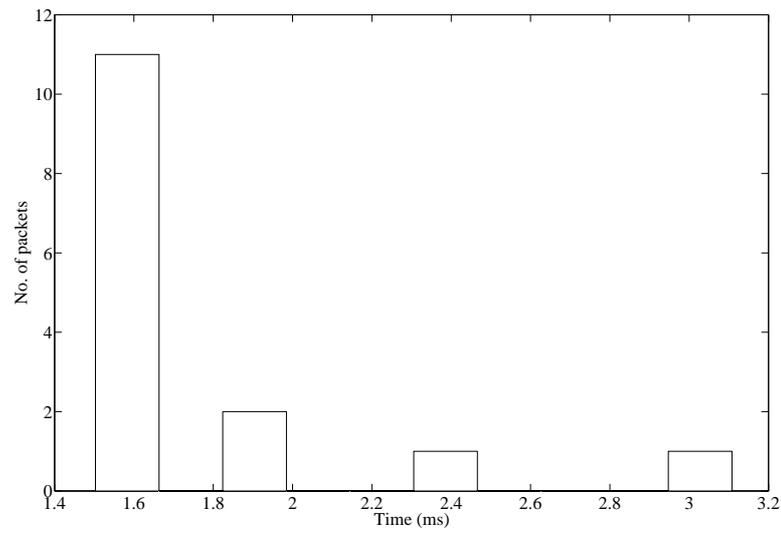
## 6.2 Web Browsing

To analyze web browsing which is based on hypertext transfer protocol (HTTP), an Apache server is installed on the server (ESA) and a website is hosted on it. The client (Surrey) browses that website via Firefox. Meanwhile packets are captured with Wireshark on both the client and server.

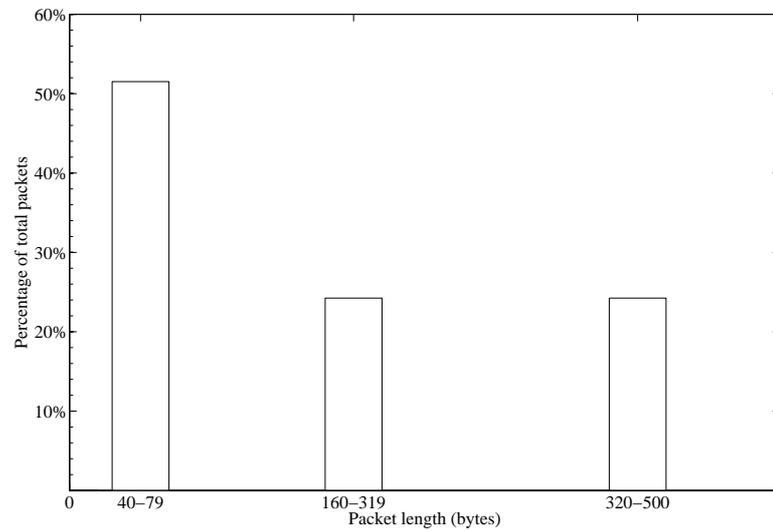
The round trip time (RTT) for the HTTP request-response pairs is shown in Figure 6. Most of the RTT values are within the stated range for a geostationary satellite, but some are higher due to the delay incurred in transferring the large objects from web server to the browser.

TCP acknowledgements for the GET requests from the client are analyzed to find their RTT. It is observed that the ACKs are coming from the performance enhancing proxy (PEP) rather than the server as inferred from the time scale in Figure 7.

The IP packet length of the packets captured during browsing the website is illustrated in Figure 8. The first bar represents the TCP ACKs, the second is HTTP responses and the third is HTTP requests. The size of the request and response data is same because each request has a single response.

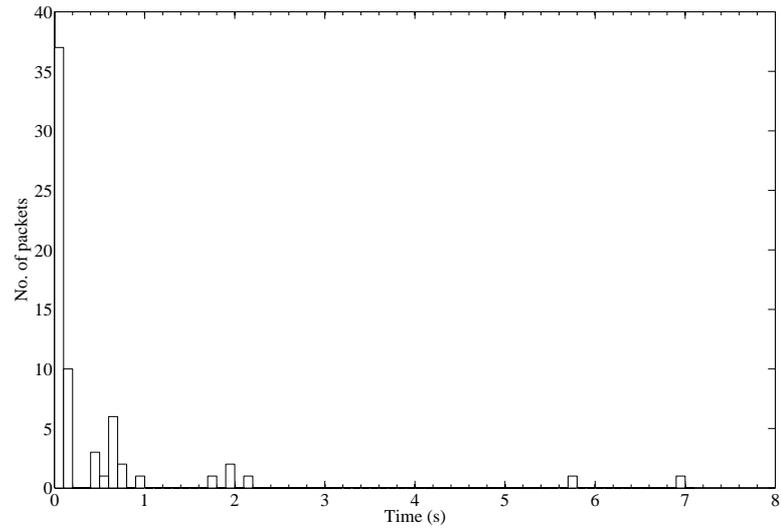


**Fig. 7.** RTT for ACK from PEP for GET requests from client



**Fig. 8.** IP packet length during browsing

The inter-arrival time of the frames during browsing is shown in Figure 9. Majority of the frames have inter-arrival times in the range of microsecond and millisecond. Some of the frames show more inter-arrival time due to the time spent in opening the browser and clicking the hyperlinks on the webpage by the user.

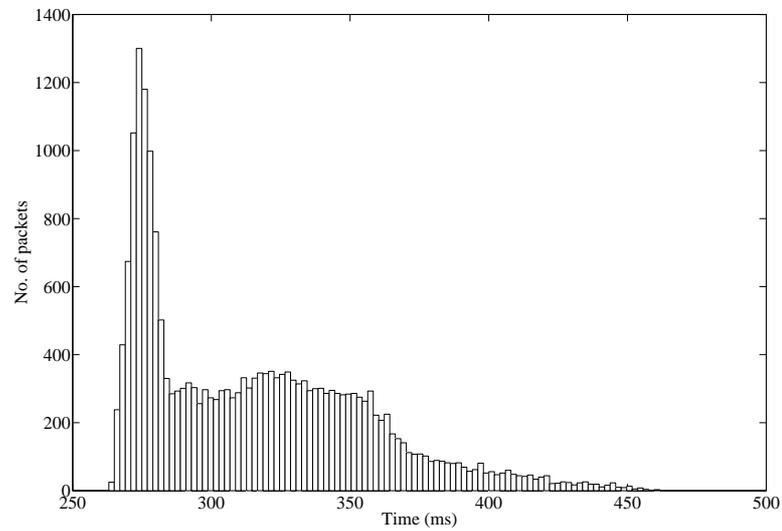


**Fig. 9.** Frame inter-arrival time during browsing

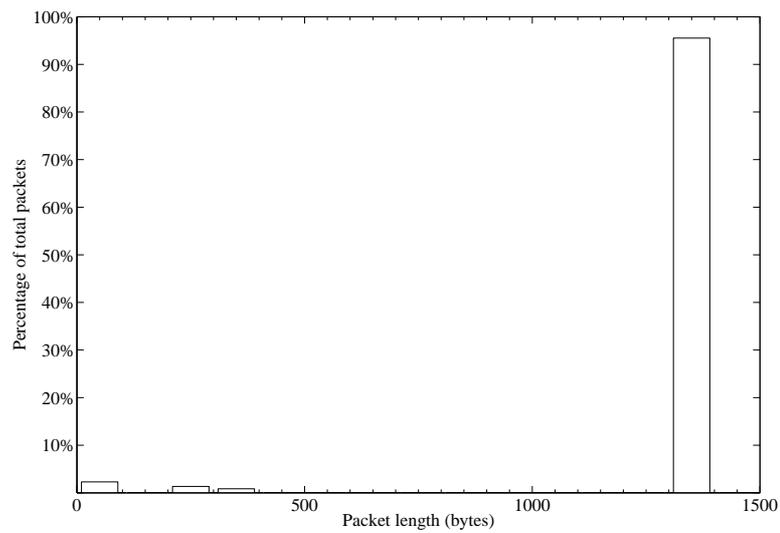
### 6.3 Video Streaming

For analysis of video streaming using user datagram protocol (UDP), a video clip has been streamed from the server (ESA) using the media player VLC and the stream is watched at the client (Surrey) with same media player.

The propagation delay for geostationary satellite is 250-270 milliseconds. The one-way delay observed between the streaming server and client is shown in



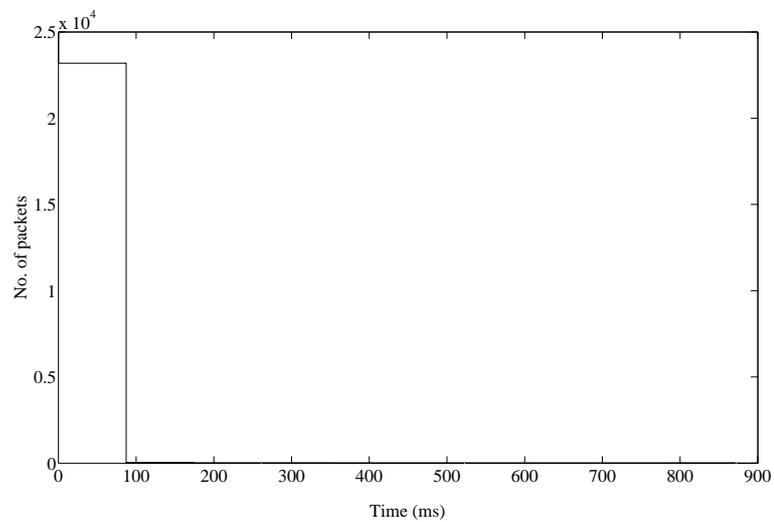
**Fig. 10.** One-way delay for video streaming



**Fig. 11.** IP packet length of streaming video packets

Figure 10. The reasons for additional delay are jitter, buffering, playout delay etc.

The IP packet length captured in streaming the video is shown in Figure 11. Most of the streaming packets are 1344 bytes in length, which is shown by a prominent bar in the histogram.



**Fig. 12.** Frame inter-arrival time for video streaming

The inter-arrival time of the frames is shown in Figure 12. As for the other two applications, file transfer and web browsing, video streaming also has inter-arrival time in microsecond and millisecond range.

Multimedia applications, like, video streaming, are sensitive to packet loss. The packet loss measured during streaming the video clip from server to client is 1.563% which is within acceptable range.

## 7 Conclusion

This paper presented the evaluation of the end-to-end QoS for different applications over a DVB-RCS satellite network. The results show the role of performance enhancing proxies (PEP) in the satellites for TCP-based applications. PEPs reduce the time of acknowledging the packets transmitted over satellites. The frame inter-arrival time for file transfer and web browsing is quite comparable while it is much less in video streaming than these two applications. It is due to the fast nature of UDP guaranteeing no reliability. More than 90% of packets in UDP based application are of same length due to no acknowledgements while, in TCP based applications, at least 30-60% of the packets are small in size and are used for acknowledgements. These acknowledgements can be further reduced by developing algorithms to reserve more bandwidth for TCP based applications over satellites.

These experiments can be tested on an IPv6 DVB-RCS satellite network testbed for comparison. More accuracy can be achieved by using GPS system for synchronization of client and server, which is more precise than the Network Time Protocol (NTP).

## Acknowledgment

The authors acknowledge the support of the European Space Agency (ESA) and the Framework 6 Network of Excellence, SatNEx II.

## References

1. VLC media player, <http://www.videolan.org/vlc>
2. Wireshark, <http://www.wireshark.org>
3. Digital Video Broadcasting(DVB), Framing structure, channel coding and modulation for 11/12 GHz satellite services (1997)
4. Digital Video Broadcasting(DVB); Interaction channel for Satellite Distribution Systems (2005)
5. Feighery, P., Scott, K., Goldsmith, D.: Using Performance Enhancing Proxies with Demand Assigned Multiple Access systems. In: Proc. IEEE Military Communications Conference MILCOM 2008. pp. 1-7 (16-19 Nov 2008)
6. Hadid, I., Gordon, D.: Apache Talking IPv6. Linux Journal (2003)

7. Mills, D.L.: Ntp: Network time protocol (ntp) version 3 specification, implementation and analysis (March 1992)
8. Skinnemoen, H., Leirvik, R., Hetland, J., Fanebust, H., Paxal, V.: Interactive IP-network via Satellite DVB-RCS. *IEEE Journal on Selected Areas in Communications* 22(3), 508–517 (April 2004)
9. Sun, Z.: *Satellite Networking: Principles and Protocols*. John Wiley & Sons (2005)