QoS Evaluation of HTTP over Satellites

Lukman Audah
Faculty of Electrical and Electronic Engineering
Universiti Tun Hussein Onn Malaysia
86400 Parit Raja, Batu Pahat, Johor, Malaysia
hanif@uthm.edu.my

Zhili Sun and Haitham Cruickshank
Centre for Communication Systems Research (CCSR)
University of Surrey
Guildford, Surrey GU2 7XH, United Kingdom
{l.audah, z.sun, h.cruickshank}@surrey.ac.uk

***Abstract*—This paper presents the studies for the end-to-end QoS of IP over integrated terrestrial and Next Generation Satellite Network (NGSN) using HTTP web application. We compare between Big-LEO and EuroSkyWay like satellites constellations for the QoS parameters (e.g. delay, loss ratio, throughput and connection duration) of request-response HTTP connections from a remote server in London and a remote client in Boston. We model the HTTP request-response with multiple connections and response files sizes variations. We create the network scenario with error model to simulate the transmission loss environment using NS-2 simulation software. A Differentiated Services (Diffserv) queue interface is placed in the terrestrial network on the server side to regulate and differentiate the traffic flows across the narrow bandwidth of the satellite links. The results showed a good performance evaluation comparison of the QoS parameters involved in the HTTP web communications across LEO and GEO satellite systems.**

Keywords-component; Quality of Service (QoS); IP over Satellite; Diffserv; HTTP Application; Integrated Network

I. INTRODUCTION

The Next Generation Satellite Network (NGSN) plays a very important role in providing ubiquitous communications across the globe. It has unique characteristics like large coverage area, fast network deployment and native broadcasting/multicasting services that extend the Internet connectivity *anywhere-anytime*. The latest standard development from European Telecommunications Standards Institute (ETSI) [1] on digital video broadcasting like DVB-S/S2 [2, 3] for the forward channel and DVB-RCS [4] on the return channel has made the satellite technology able to provide high speed Internet broadband at competitive and economical pricing rate (e.g. Tooway [5]).

The NGSN consists of integration of both terrestrial and satellite networks. Synchronize connections between the two networks are vital in order to provide optimum end-to-end QoS. The terrestrial networks have the advantage in term of technology, bandwidth and speed (e.g. high speed and low bit-error-rate of optical fibre) compared to the satellite networks that have narrow bandwidth and more prone to the transmission loss. Due to that advantage, the terrestrial network may leverage the data transfer over the satellite by adopting a control mechanism such as Diffserv [6] to regulate and differentiate the traffic flows right before being transmitted over the satellite. Contrary to the previous study on end-to-end QoS optimization of IP over satellites as in [7], we propose a Diffserv queue interface in the terrestrial network to regulate and differentiate the multiple connections between server and client. It provides scalability by simplifying the complexity functions such as traffic classification and traffic conditioning within the edge satellite network [8, 9].

Previous related studies on end-to-end QoS of IP-Diffserv [10, 11, 12] only analyzed wired or wireless terrestrial networks without integrating with the satellite networks. None has done a top-down comparison on QoS parameters for the Hypertext Transfer Protocol (HTTP) web communications between LEO and GEO satellites systems. The HTTP is designed as an Application Layer protocol within the framework of the Internet Protocol Suite (TCP/IP). It functions as request and response protocols in the client-server computing system. It uses TCP reliable connection for the host-to-host data transfer. A client that often referred as a user agent (UA) is actually a web browser or web crawler. Meanwhile, the server is the applications system running on a computer that hosts the web site. In order to establish a HTTP connection, a client submits a request message to the server. The server is identified using Uniform Resource Locator (URL) which inter-linked the hypertext resource on the Internet. Upon receiving a request from client, the server will respond by sending a response message back to the client that contain the desired resource information content such as HTML files and also the completion status information. There are two standard versions of HTTP protocols which are HTTP/1.0 and HTTP/1.1. The HTTP/1.0 uses a separate connection to the same server for every request and response function just like the File Transfer Protocol (FTP) suite while the HTTP/1.1 uses a single connection for the client and server data transfer across the Internet [13].

This paper aims to evaluate and compare the QoS parameters (i.e. delay, loss ratio, throughput and connection duration) for the HTTP web communications between integrated terrestrial-LEO and terrestrial-GEO networks. The comparison is done based on average new connections and server response files sizes variations. The NS-2 simulator software is used to simulate the internetworking scenarios for approximately one hour of simulation time. The rest of this paper is organized as follows. Section II describes the simulation configuration. Section III discusses the simulation results and analysis. Finally, section IV presents the conclusion and future works of the research.

978-0-7695-4557-8/11 $26.00 © 2011 IEEE
DOI 10.1109/CyberC.2011.38

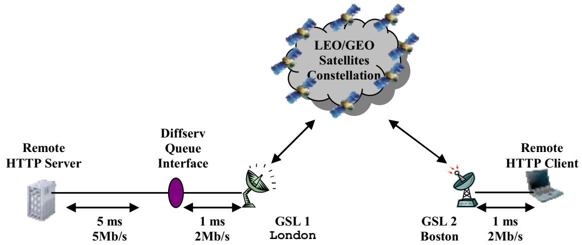


Figure 1. NS-2 simulation scenario.

II. SIMULATION CONFIGURATION

The NS-2 simulation scenario is shown in Fig.1 which consists of a remote server, a remote client, a Diffserv queue interface, two ground stations to satellite links terminals (GSL) and the LEO/GEO satellites constellation. There are two different NS-2 simulation scenarios used which are the terrestrial-LEO and terrestrial-GEO. The main difference between the two scenarios is only the satellite network parameters while the rest are exactly the same. Further details are described as follows.

A. The LEO/GEO Satellites Network

The NS-2 simulations configurations only differ in the satellites network parameters while the rest are the same for the whole simulations. We use Big-LEO constellation (i.e. 66 satellites) [14] and EuroSkyWay constellation (i.e. 5 satellites) [15] as an example of LEO and GEO satellites respectively. A remote server located in London, UK (51.53^0 N, 0^0) transmits multiple TCP-New Reno connections using HTTP web application to a remote client located in Boston, USA (42.3^0 N, 71.1^0 W). TABLE I shows the LEO and GEO parameters used throughout the simulations. Since in real world the satellites network has high transmission errors [16], therefore a random error model is introduced to simulate the characteristic. The error model produced three different bit-error-rates (BER) which are 10^{-7}, 10^{-6} and 10^{-5} for three different error scenarios.

TABLE I. LEO AND GEO SATELLITES PARAMETERS

Parameter	LEO Satellites	GEO Satellites
Altitude	780 Km	35786 Km
Planes	6	1
Satellites per plane	11	5
Inclination (degree)	86.4	0
Interplane separation (degree)	31.6	72
Seam separation (degree)	22	-
Elevation mask (degree)	8.2	8.2
Intraplane phasing	YES	YES
Interplane phasing	YES	NO
ISL per satellite	4	2
ISL bandwidth	25 Mb/s	25 Mb/s
Uplink/Downlink bandwidth	2 Mb/s	2 Mb/s
Cross-seam ISL	NO	NO
ISL latitude threshold (degree)	60	-

B. Traffic Modeling for HTTP Web Application

The application traffic used in the NS-2 simulations is based on Packmime-HTTP web object which generates the realistic synthetic web traffic [17]. We modified the average server response files sizes to be based on Pareto distribution with average discrete values of 10 Kbytes, 20 Kbytes and 30 Kbytes. Meanwhile, the average inter-arrival time for both request and response connections follow marginal distribution (e.g. a combination of modified fractional-ARIMA and Weibull distribution functions) with average new connection rates varies between 1 and 5 connection/second. We simplified the complex equations of file size and inter-arrival distributions taken from the NS-2 source codes as follows.

The current server response file size is randomly generated using Pareto distribution based on 3 average (e.g. *avg_(x)*) values which are 10 Kbytes, 20 Kbytes and 30 Kbytes respectively. Equation (1) shows the random variable of file size where *x* corresponds to the average sizes. The *RNG* corresponds to the random number generator function that generates numbers uniformly distributed between 0 and 1. The *S(x)* and *P* are the Pareto scale and shape parameters respectively. The *S(x)* in (2) is a variable based on the average files sizes in (1) while the *P* is a constant (i.e. 1.27).

$$F(x) = \frac{S(x)}{\lim\limits_{n \to 1} RNG(n)^{(1/P)}} \tag{1}$$

$$S(x) = \frac{avg\_(x) \times (P-1)}{P} \tag{2}$$

The current inter-arrival time of new HTTP connection is generated as in (3) where *p* corresponds to the fractional autoregressive integrated moving average (f-ARIMA) random distribution functions as shown in (4) and (5). The *shape* and *scale* parameters in (6) and (7) are the Weibull shape and scale variables respectively which correspond to the discrete average new connection rate (*R*) values between 1 and 5. The *A* and *C* are the sigma-epsilon and sigma-noise coefficients respectively while *B* is the f-ARIMA internal state coefficient. The *D*, *E* and *F* are the Weibull coefficients while *G* and *H* are the Gamma coefficients parameters. Further details of the parameters involved in the following equations could be read in [17].

$$I_h(t) = -\log(1-p)^{(1/shape)} \times scale \tag{3}$$

$$p = f(y(t)) = \begin{cases} 0.5 & if \quad y(t) = 0 \\ \dfrac{1 + erf\left(\dfrac{y(t)}{\sqrt{2}}\right)}{2} & if \quad y(t) > 0 \\ \dfrac{erfc\left(\dfrac{-y(t)}{\sqrt{2}}\right)}{2} & if \quad y(t) < 0 \end{cases} \tag{4}$$

$$y(t) = A \cdot f\_ARIMA(Bt) + C \tag{5}$$

$$shape = \frac{2^{\left(D+E\log\left(R/F\right)\right)}}{1+2^{\left(D+E\log\left(R/F\right)\right)}} \tag{6}$$

$$scale = \frac{1}{R \times e^{\left(\log\left(G/1+shape^{-1}\right)-H\right)}} \tag{7}$$

The TCP-New Reno segment size is set to 576 bytes (i.e. 536 bytes of payload and 40 bytes of header) with maximum congestion window size of 30 packets. The main reasons for choosing small segment size and maximum congestion window are to accommodate many HTTP web connections within the 2 Mbps of link bandwidth and also to reduce buffer overflow when the number or new connections increased. TABLE II shows the HTTP web parameters used in the simulations.

TABLE II. HTTP WEB PARAMETERS

Parameter	Value
HTTP server response file size	Model : Pareto Distribution Average (*avg_*) : 10, 20, 30 Kbytes Shape (*shape*) : 1.27
New connection inter-arrival time	Model : Marginal Distribution Average connection/second (*R*): 1, 2, 3, 4, 5
TCP type	New Reno
TCP packet size	576 bytes (536 bytes payload + 40 bytes header)

C. *Differentiated Services (Diffserv)*

Differentiated Services (Diffserv) is an Internet QoS architecture which is developed to resolve scalability problems and to provide preferential treatment to traffic flows based on class of service (CoS). The Diffserv queuing mechanism in the simulations used Random Early Detection (RED) queue and Time Sliding Window 3 Color Marker (TSW3CM) policer type which differentiate traffic flows based on 3 drop precedence (i.e. Green, Yellow and Red). Traffic flows classification will be based on the Committed Information Rate (CIR) and Peak Information Rate (PIR) which are set to 1.85 Mbps and 1.9 Mbps for the total TCP connections. This setting is to allow the maximum link utilization to be between 90% - 95% (i.e. link bandwidth of 2 Mbps).

Packets will be marked as Green if the flow rate less than CIR, Yellow if the flow rate between CIR and PIR, and Red if the flow rate more than PIR. Red marked packets will be randomly dropped first followed by Yellow and Green packets respectively only if the buffer space exceeds minimum threshold. All packets will be dropped if the buffer space exceeds maximum threshold. TABLE III shows the Diffserv queue configuration. The total buffer size of physical queue is 350 with average packet size of 576 bytes. The 3 virtual queues are virtually some fractions of the physical queue size which correspond to the minimum threshold (*minTh*) and maximum threshold (*maxTh*). Assuming that 95% of the total physical queue buffer size is used for user traffic, therefore the *maxTh* could be set to 335 packets. The *minTh* is set less than *maxTh*.

TABLE III. DIFFSERV QUEUE CONFIGURATION

Parameter	Value
Committed Information Rate (CIR)	1.85 Mbps
Peak Information Rate (PIR)	1.90 Mbps
Minimum Threshold (*minTh*)	300 packet
Maximum Threshold (*maxTh*)	335 packet
Packet Drop Probability 1 (Green)	0.01
Packet Drop Probability 2 (Yellow)	0.05
Packet Drop Probability 3 (Red)	0.10

III. RESULTS AND ANALYSIS

Each NS-2 simulation is carried out for approximately one hour of simulation time. The simulations are done 5 times (i.e. based on connection rate (*R*) values which are between 1 and 5) for each HTTP server response file size (e.g. *avg_* values which are 10, 20 and 30 Kbytes) in 3 different BER values. Therefore, the total numbers of repeated simulations are 90 times for both terrestrial-LEO and terrestrial-GEO simulation scenarios. The simulation results and analysis will be divided into 4 QoS categories which are delay, loss ratio, throughput and HTTP web connection duration. Each category refers to the IP over satellite performance metric and measured as average values per hour. The QoS parameters are calculated from NS-2 output trace using AWK programming script and then plotted on graphs using Microsoft Excel.

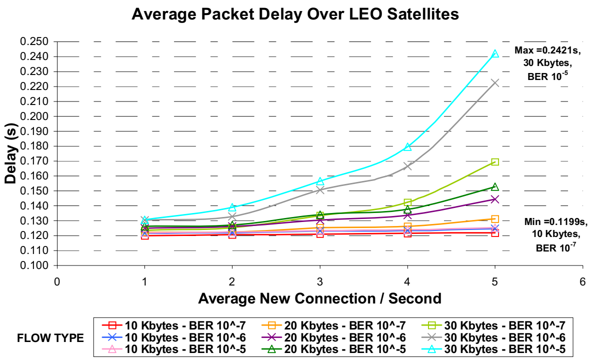
In order to get better understanding of the following figures, we use the same reference symbols and annotations. There are 9 colored lines on each graph which represent the QoS categories on 3 different HTTP server response files sizes and 3 different BER values which are 10^{-7} (i.e. "□" symbol), 10^{-6} (i.e. "x" symbol) and 10^{-5} (i.e. "Δ" symbol).

A. *Average End-to-End Packet Delay*

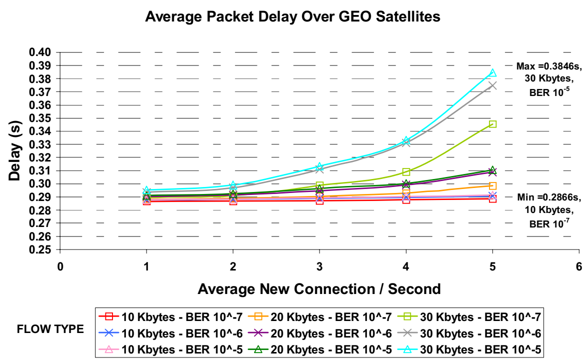
The packet delay is measured by subtracting the packet received time at the client (t_r) to the packet sending time from server (t_s). The average delay (D) is measured by summing up all packets delays and then divided by the total number of successfully received packet (P_t) at the client side as shown in the following equation.

$$D(s) = \frac{\sum_{i=1}^{i=n}\left(t_r - t_s\right)_i}{P_t} \tag{8}$$

Fig. 2 shows that the average packet delay is proportional to the increment of average new connection per second. The more new connection established per second, the higher would be the delay. In addition, the delay also increased when the BER values increased from 10^{-7} to 10^{-5} due retransmission. However, the delay values in Fig. 2 (b) are much higher than in (a) because of distinct difference in altitude between GEO and

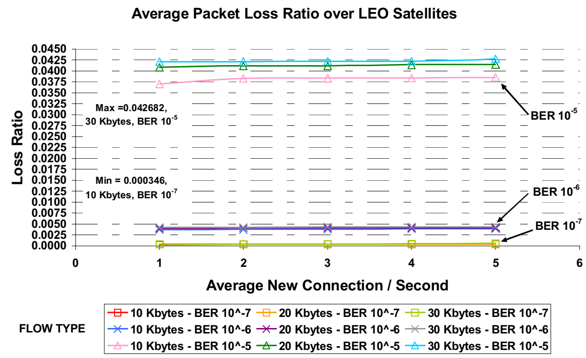


(a)

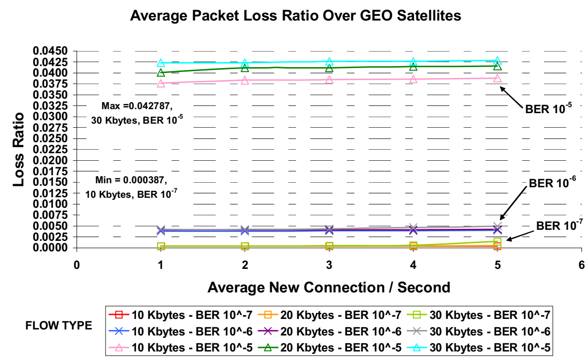


(b)

Figure 2. Average end-to-end packet delay.



(a)



(b)

Figure 3. Average end-to-end packet loss ratio.

LEO satellites. Moreover, the propagation delay over GEO satellite system is more than 250 ms [18] as opposed to the LEO satellites system which is more than 12 ms [6] depending on the hop count within the satellites network.

The delays steadily increased between 1 and 3 average new connection per second. However, after 3 average new connections per second, significant divergence could be seen between each flow of packet size with maximum average packet delay of 0.2421 second and 0.3846 second (i.e. flows with 30K bytes and BER 10^{-5}) in LEO and GEO systems respectively. In addition, the minimum average packet delays are 0.1199 second and 0.2866 second for flows with 10 Kbytes and BER 10^{-7} in LEO and GEO systems respectively. There are two main reasons that cause the delays variation which are the increment of queuing delay and the increment of packet retransmission. The queuing delay will increase when the number of incoming packets increased which will fill up the buffer space. The incoming packets of new flows keep on increasing regardless of the completion of previous flows. When the influx rate become more than the queue serving time, packets will be dropped and longer delay is needed to retransmit that packets from server to client. Besides that, the packet retransmission mainly happened because of early drop by Diffserv RED queue for the Red marked packets and also due to the packets drop in the satellite links.

B. *Average End-to-End Packet Loss Ratio*

Loss ratio (L) refers to the ratio of total packet loss (P_l) over total transmitted packet from server to client (P_s). Equation (9) shows the loss ratio calculation.

$$L = \frac{\sum_{i=1}^{i=n} (P_l)_i}{\sum_{i=1}^{i=n} (P_s)_i} \tag{9}$$

Fig. 3 shows that the average end-to-end packet loss ratio is proportional to the increment of average server response files sizes, average new connection per second and BER values. The loss ratio values for all HTTP web flows over GEO satellites are slightly more than the one in LEO system. This mainly due to the higher round-trip-time (RTT) that cause the buffer space in most queues to fill up more quickly by the influx of new connections. In addition, the Diffserv queue regulates the flows by probabilistically drop packets when buffer size exceeds minimum threshold (i.e. when packet influx rate more than queue serving time).

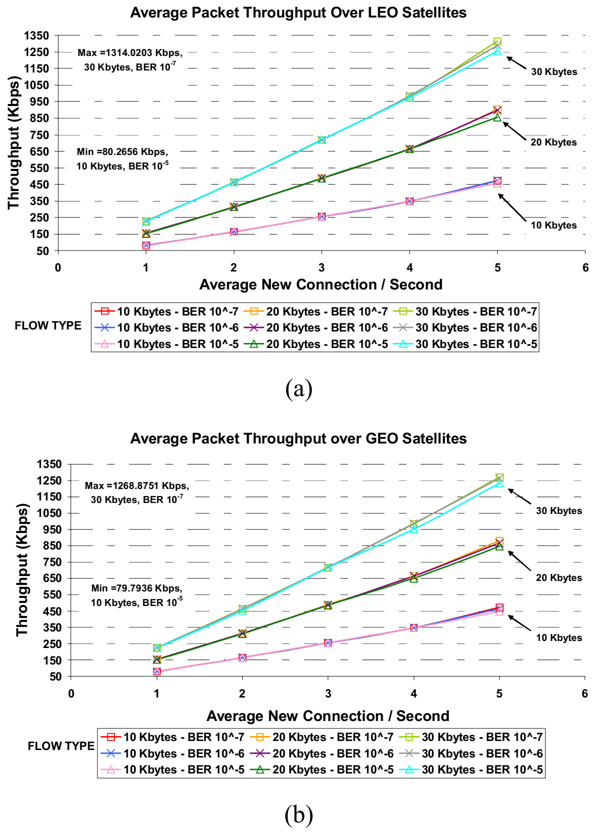


Figure 4. Average end-to-end packet throughput.

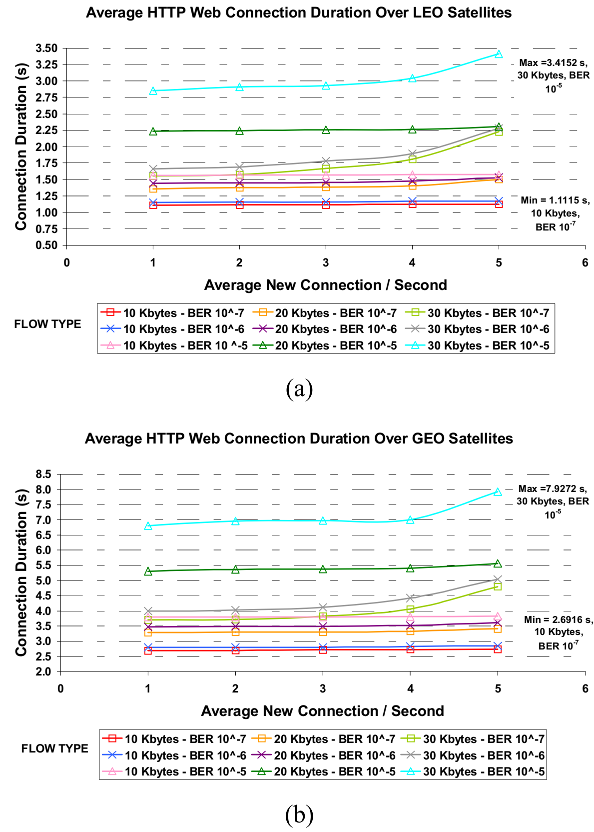


Figure 5. Average HTTP web connection duration.

Besides that, the BER in satellite network also produce significant increment in packet loss especially for BER values above 10^{-5}.

Based on the Fig. 3, the minimum average end-to-end packet loss ratio could be seen at 1 average new connection/second, 10 Kbytes average server response file size and BER of 10^{-7} which correspond to average loss ratio values of 0.000346 in terrestrial-LEO system and 0.000387 in terrestrial GEO system. The maximum average packet loss ratio values are at 5 average new connection/second, 30 Kbytes average server response file size and BER 10^{-5} which correspond to average loss ratio of 0.042682 in terrestrial-LEO system and 0.042787 in terrestrial-GEO system. The average loss rate is below 5% in worst condition due to the Diffserv QoS control and TCP reliable connection in both systems.

C. *Average End-to-End Packet Throughput*

The average end-to-end packet throughput (T) is calculated by dividing the total received packet (P_t) at the client side over the total duration of HTTP web flows. The value is then multiplied by 8 and divided by 1000 in order to get the value in Kbps. The HTTP web total duration is calculated by subtracting the receiving time of last packet at the client side (t_l) to the sending time of first packet from the server side (t_f). The total duration is slightly less than the 1 hour of simulation time because the HTTP web connections start a few seconds after the network scenario setup in NS-2 is completed.

$$T(Kbps) = \frac{\sum_{i=1}^{i=n} P_t(i)}{t_l - t_f} \times \frac{8}{1000} \tag{10}$$

The average end-to-end packet throughput concludes the previous QoS parameters because they are closely related as shown in (10). It is proportional to the total received bit variation and inverse proportional to the packet delay variation. Based on Fig. 4, the average end-to-end packet throughput is proportional to the increment of average server response file sizes and average new connection/second. The higher the average server response sizes and average new connection rate, the higher would be the P_t. However, the average packet throughput decreases as the BER increased. The P_t received at higher BER (i.e. 10^{-5}) is less than the one at lower BER (i.e. 10^{-7}) within total flow duration due to many packets losses.

D. Average HTTP Web Connection Duration

In real world, the average HTTP web connection duration could be regarded as the time conceived by the client upon sending a HTTP URL using web browser until all web contents loaded on the computer screen. The HTTP web connection duration is calculated for every completed flow within 1 hour of simulation time by subtracting the receiving time of last packet at the client side (T_l) to the sending time of first packet of a connection at the server side (T_f). The average value (C_d) is calculated by summing up all completed connection durations and then divided by the total number of completed connections (f_t) as shown in (11).

$$C_d(s) = \frac{\sum_{i=1}^{i=f_t} \left(T_l - T_f\right)_i}{f_t} \tag{11}$$

Based on Fig. 5, the average HTTP web connection duration is proportional to the increment of average HTTP server response files sizes, average new connection rates and BER values. The higher the average server response files sizes and BER, the longer would be the time needed for the connection to complete. The reason is related to the packet loss ratio which increased rapidly in higher average server response files sizes especially for higher BER values due to the retransmission of many packets loss.

IV. CONCLUSION AND FUTURE WORKS

This paper presented simulation studies to show top-down comparison between terrestrial-LEO and terrestrial-GEO networks for the end-to-end QoS performance evaluations of HTTP web application. The end-to-end QoS parameters (i.e. average packet delay, average packet loss ratio, average packet throughput and average HTTP web connection duration) are measured against the variation of average HTTP response files sizes (i.e. 10 Kbytes, 20 Kbytes, and 30 Kbytes), average new connection/second (i.e. between 1 and 5) and BER (i.e. 10^{-7}, 10^{-6} and 10^{-5}) for 1 hour of NS-2 simulation time. The average packet delay, average packet loss ratio, average packet throughput and average HTTP web connection duration are proportional to the increment of average new connection rates, average server response files sizes and BER values. Other parameters that contribute o the increment of QoS parameters are the queuing delay, buffer size and the limited link bandwidth that limit the burst of new HTTP web connections.

The future works will involve a cross-layer technique for end-to-end QoS performance enhancement. This may include the transport and network layers modifications. On the transport layer, a TCP Performance Enhancing Proxy for satellite links *satPEP* could be integrated in the system to improve TCP performance by using split connections and dynamic window resizing based on available bandwidth. This will greatly reduce the TCP packet round-trip-time (RTT) especially in terrestrial-GEO system. The *satPEP* also may have a better packet loss recovery mechanism by using Negative Acknowledgement (*NAck*). Moreover, the network layer enhancement aims to optimize the bandwidth utilization on satellite links by using load balancing technique with multiple GSL on both server and client side. This will involve multiple paths links from server to client. An admission control with Diffserv queue interface will be placed on the terrestrial network to regulate and differentiate the flow paths over the satellites based on current delay and throughput.

REFERENCES

[1] "The European Telecommunications Standards Institute," http://www.etsi.org.

[2] ETSI, "Digital Video Broadcasting (DVB) – framing structure, channel coding and modulation for 11/12 GHz satellite services," EN 300 421, August 1997.

[3] ETSI, "Digital Video Broadcasting (DVB): user guidelines for the second generation system for broadcasting, interactive services, news gathering and other broadband satellite applications (DVB-S2)," TR 102 376, February 2005.

[4] ETSI, "Digital Video Broadcasting (DVB): interaction channel for satellite distribution systems," EN 301 790, 2003.

[5] "Tooway," http://www.tooway.com.

[6] L. Audah, Z. Sun and H. Cruickshank, "End-to-end QoS of IP-Diffserv network over LEO satellite constellation," in *Proceedings of 2nd International ICST Conference on Personal Satellite Services, PSATS 2010,* Rome, Italy, February 2010, pp. 99-113.

[7] S. Kittiperachol, Z. Sun and H. Cruickshank, "Performance evaluation of on-board QoS support for multiservice applications on the integrated next generation satellite-terrestrial network," in *Proceedings of 4th International Conference on Advanced Satellite Mobile Systems, ASMS 2008*, pp 311-316.

[8] K. Nichols, S. Blake, F. Baker and D. Black, "Definition of the Differentiated Services Field (DS field) in the IPv4 and IPv6 headers," IETF Network Working Group, RFC 2474, 1998.

[9] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, "An architecture for Differentiated Services," IETF Network Working Group, RFC 2475, 1998.

[10] J. Yang, J. Ye, S. Papavassiliou and N. Ansari, "Decoupling end-to-end QoS provisioning from service provisioning at routers in the Diffserv network model," in *Proceedings of International Conference on Global Telecommunications, GLOBECOM 2004*, vol.3, no.1, pp.1605-1611.

[11] J. Yang, J. Ye, S. Papavassiliou, "Enhancing end-to-end QoS granularity in Diffserv network via service vector and explicit endpoint admission control," IEEE Journal of Communications, vol.151, no. 1, pp.77-81, 2004.

[12] L. Myounghwan, J.A. Copeland, "An adaptive end-to-end delay assurance algorithm with diffserv architecture in IEEE 802.11e/IEEE 802.16 hybrid mesh/relay networks," in *Proceedings of 18th International Conference on Computer Communications and Networks, ICCN 2009,* pp. 1-6.

[13] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach and T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1," IETF Network Working Group, RFC 2616, 1999.

[14] "Big LEO Tables," http://personal.ee.surrey.ac.uk/Personal/L.Wood/constellations/tables/tables.html.

[15] H. Cruickshank, Z. Sun, F. Carducci and A. Sanchez, "Analysis of IP voice conferencing over EuroSkyWay system," IEEE Journal of Communications, vol. 148, no. 4, pp. 202-206, 2001.

[16] M. Allman, D. Glover and L. Sanchez, "Enhancing TCP over satellite channels using standard mechanisms," RFC 2488, 1999.

[17] J. Cao, W.S. Cleveland, Y. Gao, K. Jeffay, E.D. Smith and M. Weigle, "Stochastic models for generating synthetic HTTP source traffic," in *Proceedings of 23rd Conference of the IEEE Communications Society, INFOCOM 2004*, Hong Kong, March 2004, vol. 3, pp. 1546-1557.

[18] Z.Sun, Satellite Networking: Principles and Protocols, 1st ed., John Wiley & Sons Ltd, 2005, pp. 32-34.