Journal of Communication and Computer 9 (2012) 1383-1394

Comparative Evaluation QoS of FTP over LEO and GEO Satellite Networks with Diffserv Architecture

Lukman Audah[1], Zhili Sun[2] and Haitham Cruickshank[2]

1. Faculty of Electrical & Electronic, Universiti Tun Hussein Onn Malaysia, Parit Raja 86400, Malaysia

2. Censter for Communication Systems Research, University of Surrey, Guildford, GU2 7XH Surrey, United Kingdom

Received: March 5, 2012 / Accepted: March 29, 2012 / Published: December 31, 2012.

Abstract: This paper presents studies of the end-to-end QoS of IP over integrated terrestrial and NGSN (next generation satellite network) for file transfer service using FTP. The authors compare between LEO and GEO satellites constellations for the QoS parameters (i.e., delay, jitter, loss rate and throughput) of file transfer between one server in London and a client in Boston. The authors model the file transfer with multiple connections and file size variation according to exponential and Pareto distributions respectively. The authors create the scenario with error model to simulate transmission loss environment using the NS-2 simulation software. A Diffserv (differentiated services) queue interface is placed in the server side to regulate the traffic flows across the narrow bandwidth of the satellite links. The authors compare the empirical TCP throughput traces with analytical model for validation. The results showed the performance evaluation and presented a good comparison of the QoS parameters involved in the data transfer across LEO and GEO satellites systems.

Key words: QoS, IP over satellite, Diffserv, FTP application, integrated network, TCP throughput modeling.

1. Introduction

The NGSN (next generation satellites network) plays a vital role in providing ubiquitous communications across the globe. Its unique characteristics like large coverage area, fast network deployment and native broadcasting/multicasting services extend the Internet connectivity to remote geographical area where terrestrial network is not available or not economical. With the latest standards from ETSI (European Telecommunications Standards Institute) [1] on digital video broadcasting like DVB-S/S2 [2, 3] for the forward channel and DVB-RCS [4] on the return channel, the satellite technology has been providing Internet broadband services at a competitive pricing rates, i.e., Tooway [5].

The future Internet will consists of integration of both terrestrial networks and satellite networks. Synchronize connection between the two networks is crucial in order to provide optimum end-to-end QoS (quality of service). The satellite networks are more prone to the transmission loss comparing to the terrestrial networks. In addition, the terrestrial networks have the upper hand in term of technology, bandwidth and speed (due to high speed and low bit error rate of optical fibre). The terrestrial network may leverage the data transfer over the satellite by adopting a control mechanism, i.e., Diffserv [6], to regulate and differentiate the traffic flows before being transmitted over the satellite. Unlike previous study on end-to-end QoS optimization of IP over satellites as in Ref. [7] which proposed an OBP (on-board processing) system,

Zhili Sun, Ph.D., professor, research fields: satellite network architectures, network security, TCP/IP system, VoIP. E-mail: z.sun@surrey.ac.uk.

Haitham Cruickshank, Ph.D., senior lecturer, research fields: network security, satellite network architectures, VoIP, IP conferencing over satellites. E-mail: h.cruickshank@surrey.ac.uk.

Corresponding author: Lukman Audah, Ph.D. student, research fields: QoS (quality of service), IP over satellite, integrated network routing, queuing management. E-mail: hanif@uthm.edu.my, l.audah@surrey.ac.uk.

the authors introduce Diffserv queue interface in the terrestrial network to regulate and differentiate the multiple connections between server and clients. It provides scalability by simplifying the complexity functions such as traffic classification and traffic conditioning within the terrestrial edge routers [8, 9].

Previous related studies on end-to-end QoS of IP-Diffserv [10-12] only analyzed wired/wireless terrestrial networks without integrating with the satellites networks. None has done a top-down comparison on QoS parameters for data transfer using FTP (file transfer protocol) between LEO and GEO satellites constellations. The FTP is a common Internet protocol widely used to transfer large files (mainly referred as "elephants" [13]). It is built on TCP-based client-server architecture with separation of control and data connection between the client and server. In order to make file transfer through the Internet, a client has to establish a TCP connection to the server's well-known port 21. This connection is called the control connection which will remain open for the duration of the session. Then, the server responds with three digit status code in ASCII with an optional text message (connection negotiation dialog). If the connection establishment is successful, then a second connection is opened by the server from its port 20 to the client port (which is specified in the negotiation dialog) as required to transfer a file. Due to this two-port protocol structure, the FTP is considered as out-of-band as opposed to in-band protocol like HTTP [14].

This paper aims to evaluate and compare the QoS parameters (i.e., delay, jitter, loss rate and throughput) for Internet data transfer using FTP between integrated terrestrial-LEO and terrestrial-GEO networks. The NS-2 software package is used to simulate the internetworking scenarios for approximately one hour of simulation time. The rest of this paper is organized as follows: Section 2 describes the simulation configuration; Section 3 discusses the simulation results and analysis; following the authors' previous studies on QoS over satellites as in Ref. [15], the analysis and comparison of TCP throughput results with the predicted analytical model are presented in Section 4; Finally, Sections 5 and 6 present the conclusion and future works respectively.

2. 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 link terminals (GSL) and the LEO/GEO satellites constellation. There are two different simulation scenarios used which are the terrestrial-LEO and terrestrial-GEO. Further details are described as follows.

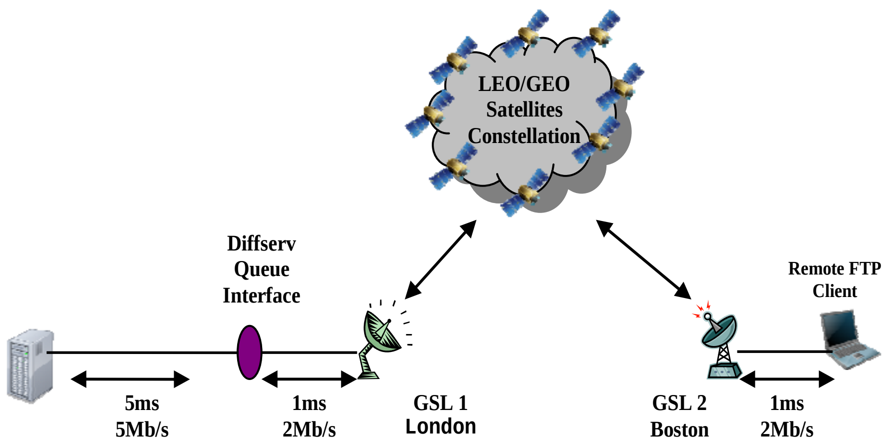


Fig. 1 Simulation scenario.

2.1 Satellite Network

The NS-2 simulations configurations only differ in the satellites network parameters. The rest are the same for the whole simulations. The authors use Big LEO (i.e., 66 satellites) [16] and Euro Skyway (i.e., 5 satellites) [17] as an example of LEO and GEO satellites constellation respectively. A remote server located in London (51.53^{0} N, 0^{o}) transmits multiple TCP connections using FTP to a remote client located in Boston (42.3^{o} N, 71.1^{o} W). Table 1 shows the LEO and GEO parameters used throughout the simulations. Since the satellites network has high transmission errors [18], a random error model is introduced to simulate the characteristic. The error model produced three different BER (bit-error-rates) which are 10^{-7}, 10^{-6} and 10^{-5} for three different error scenarios.

2.2 Data Traffic Modeling for FTP

The FTP connections vary randomly in term of average files sizes (i.e., 500 Kbytes, 1 Mbytes, 1.5 Mbytes and 2 Mbytes) and average new connection inter-arrival rate (i.e., between 1 connection/minute and 10 connection/minute) according to Pareto and exponential distributions respectively.

The TCP 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 FTP connections within the 2 Mb/s of link bandwidth and also to reduce buffer overflow when the number or new connections increased. Table 2 shows the FTP connection parameters used in the simulations.

2.3 Differentiated Services

Diffserv (differentiated services) is an Internet QoS architecture which is developed to resolve scalability problems and to provide preferential treatment to traffic flows based on CoS (class of service). The Diffserv queuing mechanism in the simulations used RED (random early detection) queue and TSW3CM (time sliding Window 3 color marker) 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 CIR (committed information rate) and PIR (peak information rate) which are set to 185 Kb/s and 190 Kb/s for a TCP connection. This setting is to allow 10 maximum average number of established TCP connections alive at a time with expected 90%-95% link utilization (i.e., link bandwidth of 2 Mb/s).

Packets will be marked as green if the flow rate within 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

Table 1 LEO and GEO satellites parameters.

Parameter	LEO satellites	GEO satellites
Altitude	780 Km	35,786 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	-

Table 2 FTP connection parameters.

Parameter	Value
FTP file size (bytes)	Model: Pareto distrbution. Average: 500 K, 1 M, 1.5 M, 2 M bytes. Shape: 1.27
New connection inter-arrival rate (connection per minute)	Model: Exponential distribution. Average: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.
TCP type	New Reno
TCP packet size	576 bytes (536 bytes payload + 40 bytes header)

only if the buffer space exceeds minimum threshold. All packets will be dropped if the buffer space exceeds maximum threshold. All physical queue sizes used in both terrestrial and satellites networks are set to 100 packets. The minimum threshold size is set 30 packets which is equivalent to the TCP maximum congestion window while the maximum threshold is set to 90 packets. The reason is to allow buffer waiting space at a time equivalent to the TCP window size agreed upon connection establishment. Data packets will randomly dropped (i.e., drop probability equal to 0.1) if the buffer size between 30 and 90 packets and all data packets will be dropped (i.e., drop probability equal to 1) if buffer size more than that. Therefore 90% of the physical queue size is allocated for the data plane while 10% for the control plane.

3. Results and Discussion

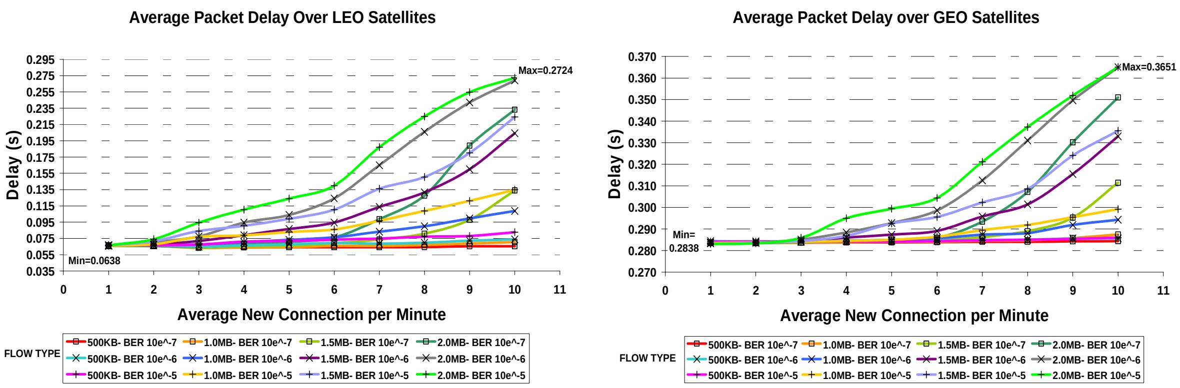
Each simulation was carried out for the duration of 1 hour of simulation time. The simulations are done 10 times (i.e., 10 average values of new connection inter-arrival time) for each FTP file size (i.e., 4 file sizes with average) in 3 different BER values. Therefore, the total numbers of repeated simulations are 240 times (i.e., for both terrestrial-LEO and terrestrial-GEO simulation scenarios). The simulation results and analysis will be divided into 4 QoS categories which are delay, jitter, loss ratio and throughput. In order to get better understanding of the following figures, the authors use the same reference symbol and annotation. There are in total of 12 colored lines on each graph which represent the QoS categories on 4 different FTP file sizes and 3 different BER values which are 10^{-7} (i.e., "□" symbol), 10^{-6} (i.e., "x" symbol) and 10^{-5} (i.e., "+" symbol).

3.1 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 in second (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:

$$D = \frac{\sum_{i=1}^{i=N} (t_r - t_s)_i}{P_t} \tag{1}$$

Fig. 2 shows that the packet delay is proportional to the increment of average new connection per minute. The more new connection established per minute, the higher would be the delay. In addition, the delay also increased when the BER values increased from 10^{-7} to 10^{-5} due to the retransmission. Obviously, the delay values in Fig. 2 (b) are much higher than in (a) because of distinct difference in altitude between GEO and LEO satellites. Moreover, the propagation delay over GEO satellite more than 250 ms [19] as opposed to the LEO satellite which is more than 12 ms [6] depending on the hop count within the satellites network.



(a) (b)

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

The delays steadily increased between 1 and 6 average new connection per minute. However, after 6 average new connections per minute, significant divergence could be seen between each flow of packet size with the maximum delay of 0.2724 and 0.3651 seconds (i.e., file size of 2 Mbytes) in LEO and GEO systems respectively. This is because of two main reasons which are the increment of queuing delay and the increment of packet retransmission. The queuing delay will increase when the number of incoming packet increase which will fill up the buffer space. The incoming packets of new flows keep on increasing regardless of the completion of previous flows. As the results, the packets incoming rates become more than the queue serving time. 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 packet drop in the satellite links.

3.2 Average End-to-end Packet Jitter

Packet jitter refers to the delay fluctuation or the delay difference between current received packet (D_c) and previous received packet (D_p). The jitter could be regarded as a vector variable because the positive value refers to the increment of current packet delay compared to the previous packet while the negative value refers to the decrement of current packet delay compared to the previous packet. Zero jitter means that the current packet delay is equal to the previous packet delay. The following equation shows the average jitter calculation:

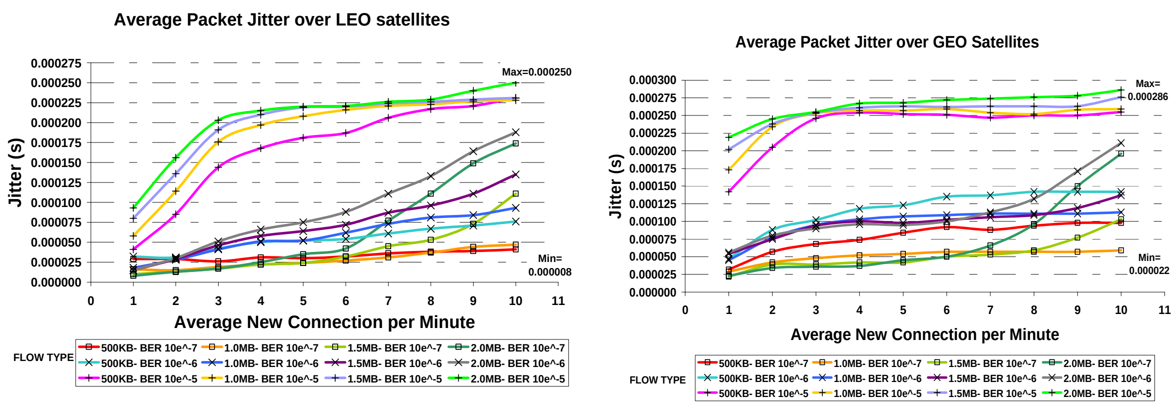
$$J = \frac{\sum_{i=1}^{i=N} \left(D_c - D_p\right)_i}{P_t - 1} \tag{2}$$

Fig. 3 shows that the average end-to-end packet jitter is proportional to the increment of average new connection per minute, average file sizes and BER. For BER values of 10^{-7} and 10^{-6}, steady increase of the average jitter could be seen between 1 and 6 of average new connection per minute and rapid increased for the subsequent connections. Higher file sizes has cause the TCP connections to remain active at longer time in order to complete the data transfer which eventually increase the influx of new connections at the queues.

As the results, jitter variation could be seen when the queuing delay and packet loss retransmission increased. However, bigger gap in jitter could be seen for the flows with BER 10^{-5} which is the worst condition. This is due to the TCP time-out as the result of too many unsuccessful received packets at the client side.

3.3 Average End-to-end Packet Loss Ratio

Average packet loss ratio (p) refers to the ratio of total packet loss (P_l) over total transmitted packet from server to client (P_s). Eq. (3) shows the loss ratio



(a) (b)

Fig. 3 Average end-to-end packet jitter.

calculation:

$$p = \frac{\sum_{i=1}^{i=N} (P_l)_i}{\sum_{i=1}^{i=N} (P_s)_i} \tag{3}$$

Fig. 4 shows that the packet loss ratio is proportional to the increment of average file sizes, average new connection per minute and BER. The loss rate values for all traffic flows over GEO satellites are slightly more than the one in LEO system. This mainly due to the higher RTT (round-trip-time) that cause the buffer space in most queues to fill up more quickly by the influx of new connections. In addition, the Diffserv regulate the flows by probabilistically drop packets when buffer size exceeds minimum threshold (i.e., influx rate > queue serving time). Besides that, the BER in satellite network also produce significant increment in loss rate especially above 10^{-6}.

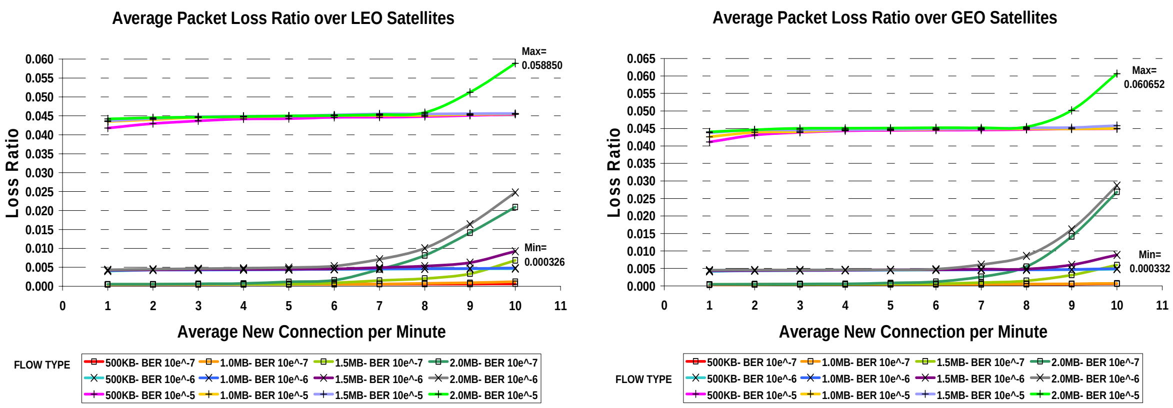
The minimum values could be seen at 1 average new connection, 500 Kbytes average file size and BER 10^{-7} which correspond to loss ratio of 0.000326 (i.e., LEO) and 0.000332 (i.e., GEO), while the maximum values are at 10 average new connection, 2 Mbytes average file size and BER 10^{-5} which correspond to loss ratio of 0.05885 (i.e., LEO) and 0.060652 (i.e., GEO). The loss rates are below 7% under worst condition due to Diffserv QoS control and TCP reliable connection.

3.4 Average End-to-end Flow Throughput

Flow throughput is calculated by dividing the total received packet bytes (P_b) over the duration of a FTP flow connection. The FTP flow duration calculated by subtracting the receiving time of last packet at the client (t_l) to the sending time of first packet of a flow at the server side (t_f). Then, the average flow throughput (B) in bit/s is calculated by summing up all completed flow throughputs and divided by the total number of completed flows (F_t) as in Eq. (4):

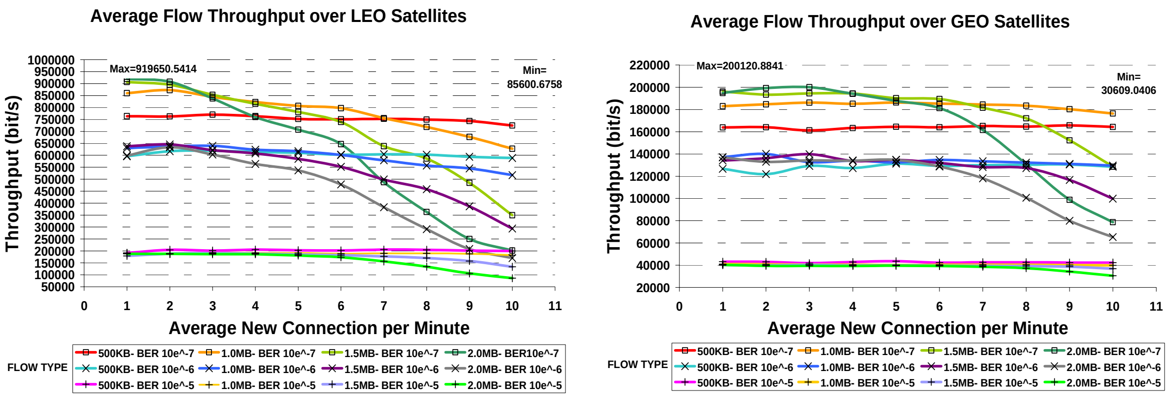
$$B = \frac{\sum_{i=1}^{i=F_t} \left(\frac{P_b \times 8}{t_l - t_f} \right)_i}{F_t} \tag{4}$$

The throughput could be regarded as the conclusion of previous QoS parameters because they are closely related as shown in Eq. (4). Based on Fig. 5, the average flow throughput is inverse proportional to the increment of average new connection per minute and BER. The more competing flows exist in network, the lower would be the average throughput seen at the client side. However, the average throughputs are proportional to the average file sizes between 1 and 2 average new connection per minute. The throughputs steadily decline on the subsequent new connections and rapid decrement soon after 6 average new connections per minute. Apart from the BER values, this is because most of the FTP flows complete before the arrival of new connections (i.e., between 1 and 2)



(a) (b)

Fig. 4 Average end-to-end packet loss ratio.



(a) (b)

Fig. 5 Average end-to-end flow throughput.

but takes long times to complete at subsequent average new connections especially after 6 average new connections due to the queuing delays and retransmission of packet loss.

4. TCP Throughput Modeling

In this section, the authors compare the TCP throughput in previous section with the mathematical modeling of TCP throughput proposed by Padhye et al. [20].

4.1 Padhye's TCP Throughput Model and Analysis

Padhye's model is a simple analytical characterization of the steady state throughput of a bulk transfer TCP flow as a function of maximum window size, loss rate, round trip time, and retransmission timeout. The studies were then validated by empirical measurements between 18 hosts scattered across the United States and Europe. Two different collective sets of data are obtained from two different measurement configurations. The first data sets are corresponded to 1 hour long of 24 TCP Reno connections, each of which the sender behaves as an "infinite source" which always has data to send and only limited by the TCP congestion control. The second data sets are corresponded to 13 pairs of TCP sender-receiver. Each pair serially initiated 100 TCP connections for 100 seconds, and was followed by a 50 seconds gap before the next connection was initiated. Based on these measurements, Padhye's model has been empirically validated and proven to be accurately predicted the TCP throughput over a significantly wider range of loss rates compared to the previous studies in Ref. [21].

Padhye concluded in Ref. [20] that the TCP throughput model in Ref. [21] was inaccurate and highly estimated the TCP throughput mainly because it predicts throughput by assuming packet losses only based of triple-duplicate acknowledgements "TD-Only". Padhye also highlighted the importance of including TCP retransmission TO (time out) behavior in the modeling perspective because this behavior commonly observed in real-time TCP throughput measurement.

The derivation of Padhye's model is based on a few main assumptions. The first assumption was that packet losses within a round (back-to-back transmission) are independent of packet losses in other round whereas packed losses within a round are correlated. The correlation of packet loss within a round assumed that if a packet loss in a round of TCP transmission, then all remaining packets transmitted until the end of the round are also lost which also has been justified previous study [22]. The second assumption was that TCP round-trip-time is independent of window size. The third assumption was that the time spent in TCP slow-start phase is negligible compared to the total duration of TCP connection. Both of second and third assumptions were also made in previous related studies in Refs. [21, 23, 24].

The throughput (B_t) in Ref. [20] is commonly presented using Eq. (5) where B_t represents the number of packets sent per unit of time regardless of their eventual fate (e.g., received or lost).

$$B_t = \lim_{t\to\infty} \frac{N_t}{t} \tag{5}$$

Eventually the long term steady state TCP throughput B(p) as a function of packet loss probability (p) can also be expressed in Eq. (6), where Y_i define the number of packets sent in the i^{th} triple-duplicate (TD) period and A_i is the duration of the period.

$$B(p) = \frac{E[Y]}{E[A]} \tag{6}$$

Then E[Y] is defined in Eq. (7) as a function of E[α] and window size E[W].

$$E[Y] = E[\alpha] + E[W] - 1 \tag{7}$$

The derivation of E[α] considers a random process of $\{\alpha_i\}_i$, where α_i is a number of packets sent in a TD period up to and including the first packet that is lost. Based on the previous assumption that packets lost in a TCP transmission round are independent of any packet lost in other round, then $\{\alpha_i\}_i$ can be regarded as a sequence of independent and identically distributed (i.i.d.) random variables. Thus E[Y] can be written as a function of loss probability (p) and window size E[W] as shown in Eq. (8).

$$E[Y] = \frac{1-p}{p} + E[W] \tag{8}$$

The complex derivation of E[W] and E[A] values in Ref. [20] as a function of packet loss probability (p), number of packet that are acknowledged by a received ACK (b) and the RTT (round-trip-time) lead to Eqs. (9) and (10) respectively.

$$E[W] = \frac{2+b}{3b} + \sqrt{\frac{8(1-p)}{3bp} + \left(\frac{2+b}{3b}\right)^2} \tag{9}$$

$$E[A] = RTT\left(\frac{2+b}{6} + \sqrt{\left(\frac{2b(1-p)}{3p} + \left(\frac{2+b}{6}\right)^2\right)} + 1\right) \tag{10}$$

Hence, from Eqs. (6), (8), (9) and (10) the TCP throughput, B(p), for the "TD-Only" model can be expressed as in Eq. (11). The model is derived without considering the maximum TCP window limitation and thus the TCP throughput values can grow toward infinity [20].

$$B(p) = \frac{\frac{1-p}{p} + \frac{2+b}{3b} + \sqrt{\left(\frac{8(1-p)}{3bp} + \left(\frac{2+b}{3b}\right)^2\right)}}{RTT\left(\frac{2+b}{6} + \sqrt{\left(\frac{2b(1-p)}{3p} + \left(\frac{2+b}{6}\right)^2\right)} + 1\right)} \tag{11}$$

The Padhye's model expands the "TD-Only" model in Eq. (11) to include both TD and TO loss indications as well as the TCP window limitation factor. The model modifies Eq. (6) and expressed the TCP throughput as in Eq. (12).

$$B(p) = \frac{E[Y] + (Q \times E[R])}{E[A] + (Q \times E[Z^{TO}])} \tag{12}$$

The E[R] and E[Z^{TO}] are the expected values of the R_i and ${Z^{TO}}_i$ variables. R_i refers to the total number of packet retransmissions in the duration of timeout sequence ${Z^{TO}}_i$. E[R] and E[Z^{TO}] variables are defined as in Eqs. (13) and (14) respectively.

$$E[R] = \frac{1}{1-p} \tag{13}$$

$$E[Z^{TO}] = TO\frac{f(p)}{1-p} \tag{14}$$

where

$$f(p) = 1 + p + 2p^2 + 4p^3 + 8p^4 + 16p^5 + 32p^6 \tag{15}$$

The derivation of Q variable can be approximated as in Eq. (16) where $\hat{Q}(w)$ is the probability that a loss occurs in a TCP window of size w is a TO.

$$Q \approx \hat{Q}(E[W]) \tag{16}$$

Finally, the complete characterization of TCP throughput B(p) is defined in Eq. (17) which is referred as the "full-model".

$$B(p) = \begin{cases} \dfrac{E_1[Y]}{E_1[A]} & E[W_u] < W_{\max} \\ \dfrac{E_2[Y]}{E_2[A]} & otherwise \end{cases} \tag{17}$$

Where

$$E_1[Y] = \frac{1-p}{p} + E[W] + \hat{Q}(E[W])\frac{1}{1-p} \quad (18)$$

$$E_2[Y] = \frac{1-p}{p} + W_{\max} + M(p) \quad (19)$$

$$E_1[A] = RTT\left(\frac{b}{2}E[W_u] + 1\right) + \hat{Q}(E[W])T_O\frac{f(p)}{1-p} \quad (20)$$

$$E_2[A] = RTT\left(\frac{b}{8}W_{\max} + \frac{1-p}{pW_{\max}} + 2\right) + N(p) \quad (21)$$

The M(p) and N(p) variables are specified in Eqs. (22) and (23) respectively.

$$M(p) = \hat{Q}(W_{\max})\frac{1}{1-p} \quad (22)$$

$$N(p) = \hat{Q}(W_{\max})T_O\frac{f(p)}{1-p} \quad (23)$$

The $E[W_u]$ is the unconstrained window size variable defined in Eq. (9) and W_{max} is the maximum TCP window size. The approximation of B(p) in Eq. (17) is shown in Eq. (24) and referred as the "approximate - model".

$$B(p) \approx \min\left(\frac{W_{\max}}{RTT}, \frac{1}{RTT\sqrt{\frac{2bp}{3}} + T_m}\right) \quad (24)$$

where

$$T_m = T_O \min\left(1, 3\sqrt{\frac{3bp}{8}}\right)p\left(1 + 32p^2\right) \quad (25)$$

4.2 Timeout and Round-trip-time Variables in NS-2

The derivation of timeout, TO and round-trip-time, RTT variables are the crucial factors in TCP throughput calculations. NS-2 follows the recommendation of Refs. [25, 26] for the mathematical derivation of both variables which has been comprehensively elaborated in Refs. [27, 28]. The C++ source codes for the TCP algorithm can be found in tcp.cc and tcp.h files of NS-2 for simulation programming references.

The NS-2 optimized the TO value to balance a tradeoff condition. A small TO value will lead to unnecessary packet retransmission while a large TO value will cause high latency of packet loss detection. The TO variable has been defined as a function on network RTT which is the time required for a data bit to travel from a source node to the destination node and travel back to the source node. The RTT may vary for each transmitted packet due to the network dynamic condition.

The smoothed (average) RTT($\bar{t}$) and RTT variation (σ_t) are computed based on the collected RTT samples which the used to compute the RTO value. Based on Ref. [25], the instantaneous smoothed RTT, RTT variation and instantaneous TO are computed using the following equations. Let t(k) be the k^{th} RTT sample collected upon receiving ACK from the receiver. Next, let $\bar{t}(k)$, $\sigma_t(k)$ and TO(k) be the values of $\bar{t}$, σ_t and TO respectively when k^{th} RTT sample is determined. Then the variables are defined as follows.

$$\bar{t}(k+1) = \alpha \times \bar{t}(k) + (1-\alpha) \times t(k+1) \quad (26)$$

$$\sigma_t(k+1) = \beta \times \sigma_t(k) + (1-\beta) \times \left|t(k+1) - \bar{t}(k+1)\right| \quad (27)$$

$$T_O(k+1) = \min\{ub, \max\{lb, [\gamma \times C(k)]\}\} \quad (28)$$

where

$$C(k+1) = \bar{t}(k+1) + 4 \times \sigma_t(k+1) \quad (29)$$

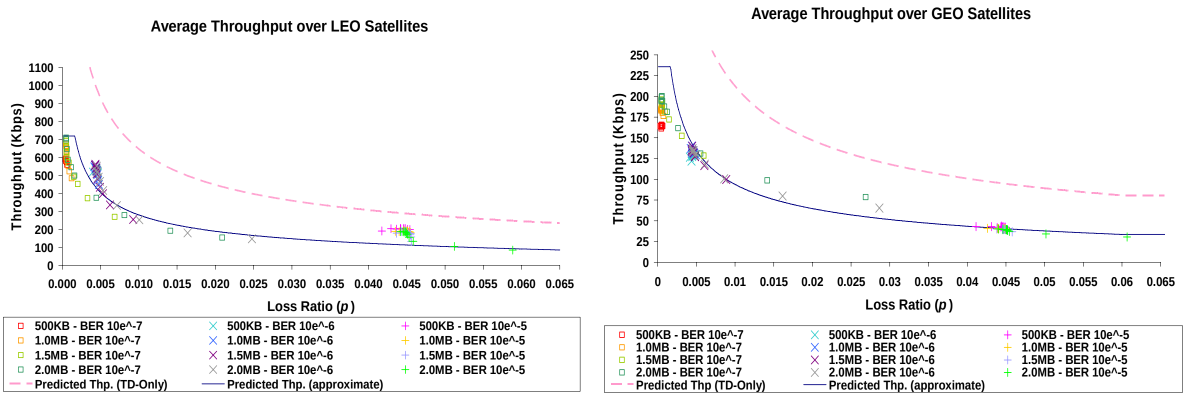
The *ub* and *lb* are the constant upper and lower bounds on the TO value. The default values for *ub* and *ul* in NS-2 are 60 second and 0.2 second respectively. The constants $\alpha \in (0,1)$ and $\beta \in (0,1)$ are usually set to 7/8 and 3/4 respectively. The variable γ is the BEB (binary exponential backoff) factor. It is initialized to 1 and doubled for every timeout event and is reset to 1 when a new ACK packet arrives.

4.3 TCP Throughput Comparison

In this subsection, the authors compare between the TCP throughput obtained from the NS-2 simulation results (section 3d)) and the Padhye's TCP approximate-model as in Eq. (24) and also TD-Only model as in Eq. (11). The authors plotted two separate graphs as in Fig. 6(a) and (b) for LEO and GEO scenarios respectively.

The RTT and TO parameters are calculated based on collected data in section 3a). Some assumptions for the parameters calculations are made as follows.

$RTT_i \approx (2 \times D_i)$: the authors assumed that the round-trip-time is approximately two times the



(a) (b)

Fig. 6 Average throughput comparison.

en-to-end delay as in Eq. (1).

$\bar{t} \approx \overline{RTT}$: the authors assumed that the average smoothed RTT, $\bar{t}$, is approximately equal to the average RTT_i as in *i*) calculated from all D_i trace data in precious section 3a).

$\sigma_t \approx \sigma\left(\sum_{i=1}^{n} RTT_i\right)$: the authors assumed that the average RTT variation, σ_t, is approximately equal to the standard deviation of all RTT_i calculated in *i*) with mean value as in *ii*).

Then, the average TO parameter is calculated using Eq. (28) based on average values approximation obtained previously. Noted that the TO in Eq. (25) is referring to the initial timeout duration as stated in Ref. [20]. Therefore, the BEB factor (γ) in Eq. (28) is set to 1. As the results, the authors obtained the average RTT and TO parameters as shown in Table 3 for LEO and GEO satellites scenarios. These values are then used to plot the predicted TCP throughput of approximate-model and TD-Only model as a function of packet loss ratio (p) as shown in Fig. 6 (a) and (b).

An important observation to be drawn from Fig. 6 is that the TD-only model is over estimated the TCP throughput on both LEO and GEO simulation scenarios especially at lower p values. This is mainly because the model did not include the effect of TCP retransmission timeout in the equation. Padhye's approximate-model is more accurate to predict the TCP throughput as most of the trace values fall within or closely to the model boundary.

Based on Fig. 6(a), the throughput for higher BER traces (e.g., 10^{-6} and 10^{-5}) are slightly above the approximate-model values for lower average TCP file sizes (e.g., short TCP connection). However, the throughput values tend to get closer or fall within the predicted boundary for higher average TCP file sizes (e.g., long TCP connection). This mainly because the instantaneous RTT, RTT variation and TO values are higher in the above mentioned conditions. The highly fluctuation of RTT values over LEO scenario is correlated to the rapid handover process between ground station to satellite and also among the satellites in the constellation. In addition, due to the lower altitude of LEO satellites, the average TCP connection duration is relatively short and eventually the average packet loss per connection is smaller than the predicted model.

Unlike Fig. 6(a), the throughput traces in Fig. 6(b) are more aligned to the approximate model. In GEO

Table 3 Key parameters for TCP throughput modeling.

Parameters	LEO	GEO
Average round-trip-time, RTT (second)	0.19246667	0.58629833
Average round-trip-time variation, σt (second)	0.04919397	0.01799251
Average timeout, TO (second)	0.38924256	0.65826836

Maximum throughput (Wmax/RTT) (Kbps)	718.25243	235.78399

satellites simulation scenario, the average RTT variation is approximately 1/3 smaller while the average RTT is approximately thrice than the one in LEO. In addition, the average TO value is almost double than the average TO in LEO. All these factors lead to the longer average TCP connection duration and eventually the average throughput per connection is relatively smaller than the predicted values in most *p* values.

5. Conclusions

This paper presented simulation studies to show top-down comparisons between terrestrial-LEO and terrestrial-GEO networks for the end-to-end QoS performance evaluations of FTP file transfers. The end-to-end QoS parameters (i.e., average delay, average jitter, average loss ratio and average flow throughput) are measured against the variation of average FTP file sizes (i.e., 500 Kbytes, 1.0 Mbytes, 1.5 Mbytes and 2.0 Mbytes), average new connection rate (i.e., between 1 and 10 connection/minute) and BER (i.e., 10^{-7}, 10^{-6} and 10^{-5}) for 1 hour of NS-2 simulation time. The average delay, jitter and loss ratio are proportional to the increment of average new connection per minute, average FTP file sizes and BER while the average flow throughput is vice-versa. Apart from the BER that significantly contribute to the increment of QoS parameters, the queuing delay, buffer size and scarce bandwidth limit the influx of new connections.

Based on the TCP throughput comparison, our hypothesis is that both TD-only model and approximate model are more suitable to predict the throughput of long TCP connection, where the steady state condition occurs for long period of time. The Markovian process assumption that packets lost in a TCP transmission round are independent of any packet lost in other round seems inaccurately predicted the throughput in some cases of our simulation studies especially in LEO network scenario. In our simulation studies, the random variables of inter-arrival time and files size of the TCP connection follow the Exponential and Pareto distributions respectively. New connection with variable file size starts randomly without waiting for the previous connection to finish. If many connections take a lot of time to finish, then the accumulative bandwidth usage may grow even bigger. At a time, there will be series of congestions or packet losses on each active TCP connection and eventually degrades the global TCP throughput. Furthermore, the effect link handover and packets buffering processes also significantly contribute to the variation of end-to-end TCP throughput. In addition to the above mentioned improvement factors, a more precise throughput calculation can be obtained if the analytical model includes the effect of fast recovery and fast retransmit mechanisms as these are the key elements of TCP reliable data transmission.

There still works remain for further studies. One of these is to achieve maximizing the bandwidth utilization on the satellite links by using load balancing method with multiple GSL on both server and client sides. This will involve multiple paths links from server to client. An admission control with Diffserv queue interface will be placed on the server side to regulate and control the flows paths over the satellites based on the current delay and throughput. This method will optimize the end-to-end QoS of multiservice applications like HTTP, FTP, video streaming and VoIP over the satellite links.

Acknowledgements

The authors are pleased to acknowledge the support of UK-ESPRC and MONET project.

References

[1] The European Telecommunications Standards Institute (ETSI) Official Website, http://www.etsi.org .

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

[3] European Standard (Telecommunication Series), 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, 2005.

[4] European Standard (Telecommunication Series), Digital Video Broadcasting (DVB): interaction channel for satellite distribution systems, EN 301 790, 2003.

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

[6] L. Audah, Z. Sun, 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, 2010.

[7] S. Kittiperachol, Z. Sun, 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, Bologna, Italy, 2008.

[8] K. Nichols, S. Blake, F. Baker, D. Black, Definition of the differentiated services field (DS field) in the IPv4 and IPv6 headers, IETF Network Working Group, 1998.

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

[10] J. Yang, J. Ye, S. Papavassiliou, 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, Dallas, Texas, USA, 2004.

[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 151 (2004) 77-81.

[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, San Francisco, California, USA, 2009.

[13] K. Papaguannaki, N. Taft, S. Bhattacharyya, P. Thiran, K. Salamatian, C. Diot, A pragmatic definition of elephants in internet backbone traffic, in: Proceedings of 2nd ACM Worshop on Internet Measurement, IMW 2002, Marseille, France, 2002.

[14] J. Postel, J. Reynolds, File Transfer Protocol (FTP), Request for Comments (RFC) 959, 1985.

[15] L. Audah, Z. Sun, H. Cruickshank, End-to-end QoS evaluation of IP over LEO/GEO satellites constellations for FTP, in: Proceedings of 5th International Conference on Signal Processing and Communication Systems, ICSPCS2011, Honolulu, USA, 2011.

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

[17] H. Cruickshank, Z. Sun, F. Carducci, A. Sanchez, Analysis of IP voice conferencing over Euro Skyway system, IEEE Journal of Communications 148 (2001) 202-206.

[18] M. Allman, D. Glover, L. Sanchez, Enhancing TCP over satellite channels using standard mechanisms, Request For Comments (RFC) 2488, 1999.

[19] Z. Sun, Satellite Networking: Principles and Protocols, John Wiley & Sons Ltd, 2005.

[20] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, Modeling TCP throughput: A simple model and its emperical validation, in: Proceedings of the ACM SIGCOMM'98 Conference on Applications, Technologies, Architectures and Protocols for Computer Communication, Vancouver, Canada, 1998.

[21] M. Mathis, J. Semke, J. Mahdavi, T. Ott, The macroscopic behavior of the TCP congestion avoidance algorithm, Journal of Computer Communication Review (CCR) 27(1997) 67-82.

[22] J. Bolot, A.V. Garcia, Control mechanisms for packet audio in the Internet, in: Proceedings IEEE Infocom96, San Francisci, California, USA, 1996.

[23] J. Mahdavi, S. Floyd, TCP-friendly unicast rate-based flow control, Technical note sent to the end2end-interest mailing list, 1997.

[24] T. Ott, J. Kemperman, M. Mathis, The stationary behavior of ideal TCP congestion avoidance, Technical note on TCP, 1996.

[25] V. Paxon, M. Allman, Computing TCP's retransmission timer, IETF Network Working Group, Request For Comments (RFC) 2988, 2000.

[26] P. Karn, C. Partridge, Improving round-trip time estimates in reliable transport protocols, in: Proceedings of ACM SIGCOMM'87 workshop on Frontiers in Computer Communications Technology, Stowe, USA, 1987.

[27] T. Issariyakul, E. Hossain, Introduction to Network Simulator NS2, Springer Science+Business Media, New York, 2009.

[28] K. Fall, K. Varadhan, The ns manual, The VINT Project, 2008, http://www.isi.edu/nsnam/ns/ns-documentation. html.