Fixed-point approximations for TCP performance over bandwidth on demand GEO satellite links

M. Karaliopoulos, R. Tafazolli and B. Evans
Centre of Communication Systems Research
University of Surrey
Guildford, GU2 7XH, United Kingdom

Abstract– **We investigate the use of fixed-point methods for predicting the performance of multiple TCP flows sharing geostationary satellite links. The problem formulation is general in that it can address both error-free and error-prone links, proxy mechanisms such as split-TCP connections and account for asymmetry and different satellite network configurations. We apply the method in the specific context of bandwidth on demand (BoD) satellite links. The analytical approximations show good agreement with simulation results, although they tend to be optimistic when the link is not saturated. The main constraint upon the method applicability is the limited availability of analytical models for the MAC-induced packet delay under non-homogeneous load and prioritization mechanisms.**

Index terms– **Bandwidth on Demand, GEO satellite networks, multiple access, TCP, fixed-point**

I. Introduction

Satellite networks will form part of the future global network infrastructure. One of the key issues for their design and commercial success is the achieved level of integration with the terrestrial networks. In the light of the IP dominance in current data networking, broadband satellite networks can be regarded as subnets over which TCP/IP has to be optimally supported. In this context, analytical methods for studying the behavior of TCP/IP flows over satellite networks are valuable.

Two main trends have emerged during the last years with regard to the analytical modeling of TCP/IP networks. Processor-sharing models [1]-[3] provide effectively an abstraction of the TCP-controlled bandwidth sharing that is performed in individual links. With the partial exception of [1], actual engineering details of the protocol are ignored. The models address the behavior of the protocol at the flow-/connection-level, without providing insight to the packet-level dynamics of the traffic such as buffer queues and losses. The most attractive property of PS-based models is their *insensitivity property*: the model outcome is not dependent on the actual distribution of connection arrivals or transferred file sizes but only on their mean. This property enables the generalization of solutions derived under Markovian theory to the more realistic cases of power-tail distributions, which are reported in literature for TCP traffic features.

In our study, we rather focus on the second approach that makes use of fixed-point methods. Contrary to the PS-based models, fixed point-methods feature separate, explicit models for the TCP source and the resource (network element-router) that are coupled with each-other: the TCP send rate is an explicit function of the loss and delay experienced by the connection in the network nodes [4]-[6], which are in turn a function of the TCP send rates via the resource models. Classical queuing models such as the M/D/1/K or the M/M/1/K model in [7]-[9], as well as Active Queue Management (AQM) models [10], [11] have been investigated as resource models. In this paper, we describe a fixed-point method for analyzing TCP performance over satellite networks. Our starting point is the vector-based representation in [7] addressing terrestrial wired networks. We generalize their formulation regarding two aspects that have to be considered when addressing satellite networks, although they are not limited to them, i.e. path and/or bandwidth asymmetry and the existence of broadcast and MAC-shared links.

The paper is organized as follows. Section II describes our generic method and its application to the specific instance of Bandwidth on Demand (BoD) satellite links. In section III we outline the analytical models used at the TCP source and satellite link level, whereas in section IV we evaluate the method via ns2 simulations and discuss the simulation results. We conclude our study in section V with some comments on the method limitations.

II. Method Description

Let L be the set of network links and R be the set of routes (paths) of the TCP connections, corresponding to subsets r_i of L. If M and N are the cardinalities of the two sets, i.e. $M = |L|$ and $N = |R|$, then we can define two N x M routing matrices $\overline{\mathbf{A}}_F$ and $\overline{\mathbf{A}}_R$, one for the forward path of the connection and one for the reverse (ACK packets path). Their elements a_{ij}^F, a_{ij}^R equal unity when the link j, $j \in L$ is included in the forward/reverse connection path i respectively, $i \in R$, and 0 otherwise:

$$a_{ij}^{F(R)} = \begin{cases} 1 & if\ \ j \in r_i \\ 0 & if\ \ j \notin r_i \end{cases} \qquad (1)$$

We also define the vectors $\mathbf{c} = (c_j)$ and $\mathbf{d} = (d_j)$ to denote the capacities of the network links and the one-way propagation delays related to them, respectively. Then the round-trip-times of the N paths $\mathbf{t} = (t_i)$ can be written as a function of queuing delays at the output buffers of the links $\mathbf{dq} = (dq_j)$ and the component of total path delay that is related to the propagation delays $\mathbf{t_p} = (t_i^p)$ as

$$\mathbf{t} = \left(\overline{\mathbf{A_F}} + \overline{\mathbf{A_R}}\right)\cdot(\mathbf{d} + \mathbf{dq}) = \mathbf{t_p} + \left(\overline{\mathbf{A_F}} + \overline{\mathbf{A_R}}\right)\cdot \mathbf{dq} \qquad (2)$$

If n_i is the number of connections sharing the same path *i*, w_i the average window sizes (at path level) and m_i^{TCP}, m_i^{ACK} the TCP maximum segment size and the ACK packet size respectively in path *i*, the total send rate in the forward and the reverse path are given by

$$\mathbf{s_F} = \mathbf{n}.\cdot \mathbf{SR}\left(\mathbf{W}^{\max}, \mathbf{m}_{\mathbf{TCP}}, \mathbf{p}, \mathbf{t}\right) \tag{3}$$

$$\mathbf{s_R} = \mathbf{n}.\cdot \mathbf{SR}\left(\mathbf{W}^{\max}, \mathbf{m}_{\mathbf{TCP}}, \mathbf{p}, \mathbf{t}\right).\cdot(\mathbf{1}-\mathbf{p}).\cdot\mathbf{u}.\cdot\frac{\mathbf{m}_{\mathbf{ACK}}}{\mathbf{m}_{\mathbf{TCP}}} \tag{4}$$

where the notations x.·y and x./y, where x and y are vectors, denote element-wise multiplication/division respectively. The vector **p** is defined in (9) and **u** accounts for the reverse path load reduction due to delayed ACKs

$$0.5 \le u_i \le 1 \quad i \in R$$

$SR_i(W_i^{\max}, m_i^{TCP}, p_i, t_i)$ denotes the mean send-rate of the connections on path *i* (see section III.B).

The offered load upon the network queues[1] is then written

$$\mathbf{b} = \mathbf{s}_\mathbf{F}^\mathbf{T}\cdot\overline{\mathbf{V_F}} + \mathbf{s}_\mathbf{R}^\mathbf{T}\cdot\overline{\mathbf{V_R}} \tag{5}$$

where the elements of the *N x M* matrices $\mathbf{V_F}, \mathbf{V_R}$ are given by

$$v_{ij}^{F,R} = \begin{cases} \prod\limits_{j' \in r_i : pos(j',i) < pos(j,i)} \left(1 - q_{j'}\right) & if\ j \in r_i \\ 0 & otherwise \end{cases} \tag{6}$$

In (6), $pos(j,i)$ yields the position of the queue (link) *j* within path *i* and q_j is the aggregate loss probability due to both congestion and link errors over the respective link.

The assumption is that

$$q_k = 0\,,\ pos(k,i) = 1 \ \ for\ k : pos(k,i) < pos(m,i)\, \forall\, m \in r_i, m \neq k$$

in other words, there is no TCP data loss at the sender parts of the connections at the end hosts, whose queues are the first ones on the connections' paths.

In general, losses and delays may occur at any link on the connection's path:

$$dq_j = f_d^j(b_j; c_j) \tag{7}$$

and
$$q_j = q_j^c + q_j^l = f_l^j(b_j; c_j) + q_j^l \tag{8}$$

where f_d^j and f_l^j are functions (formulas) yielding the delay/loss suffered at each link *j* of capacity c_j, when load b_j is offered to it. The term q_j^l is the loss probability related to link errors. In order to make the analysis feasible, the assumption of independent losses at each link is invoked, so that, if **1** is a vector of appropriate size (here: N) whose components are all equal to 1, the path losses experienced by the TCP senders on the connection paths may be written

$$\mathbf{p} = \mathbf{1} - \overline{\mathbf{A_F}}\cdot\left(\mathbf{1} - \mathbf{q}\right). \tag{9}$$

The TCP window is then dictated by the queuing losses (q_j^c) and the link errors (q_j^l) encountered at each link on the connection path and the limitations set by the buffer availability at the receiver peer or the application, $\mathbf{W}^{\max}$ (3)-(4).

The medium access controlled links need particular attention in order to be accommodated in this framework: with regard to queuing delay/loss they can be regarded as the distributed equivalent of a single conventional multiplexer, i.e. as a single distributed switch. On the contrary, with regard to link errors a shared link corresponds to many point-to-point links due to the physical separation of the sender-receiver pairs.

One way to expand the described vector representation so that it can accommodate MAC controlled links is to slightly modify (6), (8) and (9). For the MAC-shared link j^s, (8) and (9), under realistically small loss rates, may be rewritten as:

$$q_{j^s} = q_{j^s}^c = f_l^{j^s}(b; c) \tag{10}$$

and
$$\mathbf{p} = \mathbf{1} - \overline{\mathbf{A_F}}\cdot\left(\mathbf{1} - \mathbf{q}\right) + \mathbf{p^l} \tag{11}$$

respectively and the factors accounting for the successive traffic thinning on its path are written

$$v_{ij}^{F,R} = \begin{cases} \prod\limits_{j' \in r_i : pos(j',i) < pos(j,i)} \left(1 - q_{j'}\right) & if\ j \in r_i \cap pos(j,i) \le pos(j^s,i) \\ \prod\limits_{j' \in r_i : pos(j',i) < pos(j,i)} \left(1 - q_{j'}\right) - p_i^l & if\ j \in r_i \cap pos(j,i) > pos(j^s,i) \\ 0 & otherwise. \end{cases}$$

In other words, the link error probabilities at the shared link are separated from the link congestion losses and are added at connection (i.e. path) level as separate terms. These adaptations are applicable when there is a single MAC-shared link on the connections' paths or, equivalently, when other MAC-shared links, e.g. Ethernet network at the TCP server/client side, do not introduce link errors.

A. Split connections

To allow the fixed-point method to account for the split-TCP mechanism, the elements a_{ij}^F, a_{ij}^R of the new routing matrices $\overline{\mathbf{A}}_F$ and $\overline{\mathbf{A}}_R$ can be defined as:

$$\mathbf{A_F}(:,j) = \mathbf{A_R}(:,j) = \delta_{jk}\cdot\mathbf{1} \tag{12}$$

where *k* is the index of the satellite link, δ_{jk} is the Kronecker Delta and the elements w_i and $W_i^{\max}$ are replaced by the respective values in the satellite portion of the connection. The assumption behind this modification is that the satellite component of the connection determines the actual connection throughput. In general, its validity is dependent on the relative values of the actual throughputs achieved by the individual connection components –hence on all those factors affecting the throughput, such as RTT, delay/losses, enabled options in each of the connection components [12]. It holds as far as the throughput of the terrestrial component of the split connection is at least such that it does not let the satellite component starve.

[1] a single network node may be related to more than one queues, one per output link

B. Application to bottleneck BoD-shared links

We consider in the following the application of this generic formulation in the context of links shared according to BoD mechanisms. The latter are deemed mandatory in satellite networks for achieving a trade-off between resource utilization and traffic handling efficiency, in particular under non-aggregate, bursty TCP traffic. We assume that the bottleneck link is the shared satellite link, neglecting the delays incurring at the other links. We also investigate cases featuring no loss at the intermediate nodes; the control signal for TCP in our context is the delay experienced at the MAC level of the shared satellite link. In comparison with the generic problem formulation

$$dq_j = dq_{sat} \cdot \delta_{jk} \tag{13}$$

and $$\overline{\mathbf{A_F}}(:,k) = \overline{\mathbf{A_R}}(:,k) = \mathbf{1}$$

If we replace the shared link index *k* with *sat,* (7) yields

$$dq_{sat} = f_d^{sat}\left(\sum_{i=1}^{N} SR_i\left(W_i^{\max}, n_i, p_i, t_i\right)\cdot\left(1+\frac{m_{ACK}^{i}\cdot u_i}{m_{TCP}^{i}}\right), c_{sat}\right) \tag{14}$$

hence, the MAC delay at the satellite link as well as the total load offered to the satellite link b_{sat} are obtained via solution of the fixed-point equation (14). The missing elements in these equations are the models/formulas that relate the MAC access delay $dq_{sat,}$ at the shared satellite link of bandwidth c_{sat} with the aggregate send rate of TCP sources. This is the subject of the following section.

III. TCP Source and Resource Models

A. The BoD-shared link model

There is a bulk of papers on the analysis of packet satellite MAC protocols – the interested reader is referred to [13] for a comprehensive summary. Their applicability to the fixed-point approximations we describe depends on the actual treatment of TCP traffic in the satellite network and no strict rules apply there. The resource (bandwidth) allocation mechanisms may differ depending on factors such as:

- the traffic aggregation level, namely the number of connections per MAC protocol entity: Higher levels of aggregation render reservation protocols more relevant, whereas at low levels of traffic these protocols penalize resource utilization
- type of connections: connections may be short ("mice") or longer ("elephants"). Short connections result in more dynamic load variation for the same average number of connections, hence they favor dynamic reservation protocols
- the satellite network QoS policy: an operator has to resolve a trade-off between the QoS guarantees provided to the TCP flows and the resource utilization efficiency
- the loading of the network: adaptive protocols might alter the way they treat TCP traffic according to the resource availability

In the following, we are going to focus on dynamic assignment protocols, where the resource allocation is made in response to requests corresponding to the instantaneous traffic demand (buffer occupation) or is distributed, when spare, among active terminals. These types of protocols are more suited to low levels of traffic aggregation, corresponding to end-users for example, but they are also a possible approach for higher levels of traffic aggregation, in particular under less strict network QoS commitments and "mice" TCP connections. In this context, the most relevant analytical study we are aware of is the analysis provided in [14] for a member of the CFDAMA protocol family, the one with pre-assigned request slots.

The load produced by each terminal follows the Poisson distribution with frame arrival rate λ per time slot. The requests are processed by an on-board scheduler. The frame length is larger than the one hop-delay *R* (measured in slots). The basic analysis provided therein first solves a fixed-point equation for the average percentage of frame slots that are demand-assigned and then estimates the MAC delay in terms of long series of summed terms, resulting in a high computational burden. However the authors exploit the dependence of the protocol performance on the terminal population and suggest a much simpler, closed-form, approximation for the case of low terminal population *N*. The latter is quantified via the normalized terminal population $\eta_T = \frac{N}{R}$. For $\imath_T << 1$, the CFDAMA scheme was modeled as an M/G/1 queue with a geometric service discipline. The average access delay *T* was derived to be

$$T = dq_{sat} = N + \frac{\lambda\cdot N\cdot(2\cdot N-1)}{2\cdot(1-\lambda\cdot N)} \tag{15}$$

B. The TCP source model

1) Link errors: Analytical models for TCP have addressed the two main metrics that are of interest for TCP connections, TCP throughput and TCP latency. Expressions for throughput are provided in [4], the *square root p formula*, and [5], hereafter called *PFTK formula* from the initials of the paper's authors. Throughput becomes more relevant for long transfers, where the impact of the Slow Start phase is less significant. The implicit assumption in the derivation of formulas is that the connections are of infinite duration (persistent TCP connections). The main improvement of the PFTK formula over the simpler square root p formula is the modeling of timeouts, which allows it to be more accurate at higher error rates when timeouts are more probable. Neither of the two formulas addresses the Slow Start or the Fast Recovery phases of the TCP congestion control algorithm.

The TCP latency, namely the time required for the completion of a TCP transfer, is modeled in [6]. The model, hereafter called *CSA formula* from the authors' initials, accounts for the Slow Start phase and assumes finite transfer size, i.e. the size of the transfer (in bytes/segments) is one of the explicit inputs to the model. The model is better suited to small ("mice") transfers that spend a considerable amount of their lifetime in Slow Start.

2) Error free links: In the case of error-free links, the steady-state throughput expression is simplified to (16)

$$Thr = \frac{W_{\max} \cdot MSS}{RTT_{eff}} = \frac{W_{\max} \cdot MSS}{RTT + 2 \cdot dq_{sat}} \qquad (16)$$

The respective expression for the latency can be derived from the CSA formula for low enough error/loss probability or directly from the analysis in [15].

IV. Method Validation

Simulations with ns2 simulator were performed in order to assess the accuracy of the method predictions. Our BoD-specific extensions to the simulator are described in [16]. The simulation scenario is depicted in Fig. 1. Periodically, and as long as new packets arrive at their queues, the *BoD entities*, physically located in satellite terminals and serving one or more data flows, submit requests to a *BoD Controller* residing on-board the satellite. The Controller takes requests into consideration, builds the burst time plan (BTP) and broadcasts it to the BoD entities. Both functions are repeated with a period equal to an integer number of TDMA frames (*resource allocation period*). Requests are submitted in pre-assigned slots.

The BTP corresponding to requests submitted within a certain allocation period is activated by the Entities after an integer number of resource allocation periods (*system response time*), allowing for the propagation and processing delays at the terminals and the Controller. Requests that cannot be satisfied in the corresponding allocation period are stored and served with priority in the next resource allocation period. Spare capacity (*free slots*) is distributed prioritizing entities that have not submitted a request in the respective resource allocation period on a round-robin basis. A more detailed description of the actual BoD implementation is given in [17]. For the results reported subsequently, the link capacity was 2.048Mb/s, the resource allocation period (RAP) was set to 4 TDMA frames of 128 slots and 24ms time duration each, yielding a system response time of 5 RAPs.

TCP NewReno has been taken as a reference for the simulations with a TCP segment size of 526 bytes and the timestamp option activated. Our first set of simulations addresses persistent TCP connections. We compare analytical with simulation results for both one-to-one mapping between TCP connection-BoD entity and the more general case, where more than one TCP connections are handled by a single BoD entity. The method is assessed assuming both error-prone and error-free satellite links. The second simulation set investigates the method applicability to the finite transfer size case. The focus in all simulation runs is on mean connection throughput for persistent TCP flows and TCP latency, when finite duration flows are considered.

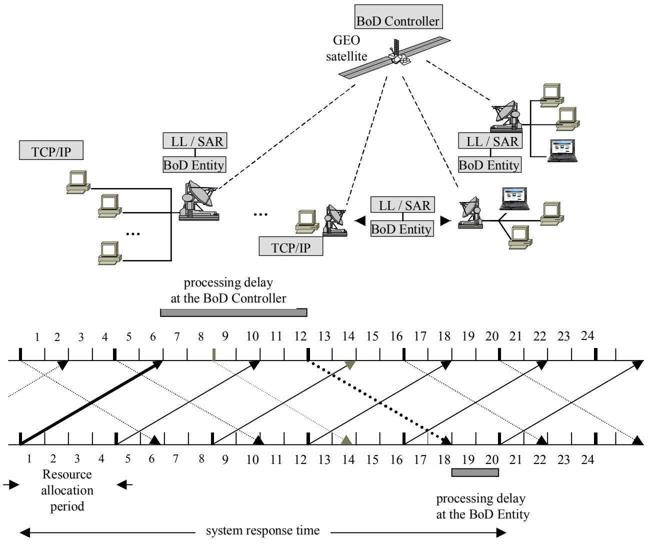

Fig. 1. The simulated network and the BoD process evolution with time

A. *Infinite duration flows (persistent connections)*

1) Error free links – single TCP connection per BoD entity: Fig. 2 compares the throughput predicted by the analytical method against the simulation results for different TCP maximum buffer size values. There is close agreement for high link loads, where the TCP throughput is limited by the availability of link capacity, i.e. the satellite link is the actual bottleneck. The accuracy of the method is less at moderate/low load and improves when the $W_{\max}$ constraint becomes active (8kB and 16kB). The deviation of the analytical estimates from the simulations results is due to the TCP burstiness. For low load Eq. (16), derived under the assumption of Poisson arrivals, predicts a much lower delay than the one actually suffered from TCP segments/ACK packets, resulting in higher estimates about TCP throughput.

The accuracy of the method increases again at very low load, when few connections with small W_{max} values share the satellite link. The access delay approaches zero and the effective round-trip time of the TCP connection almost coincides with the propagation delay. Note that the agreement is good for W_{max} equal to 8 and 16kB but it is not for 32kB (and above). In that case, the connection is aggressive enough to give rise to non-negligible access delays and generate mismatch between the model estimate in (15) and the real access delay experienced by TCP segments.

2) Error free links – multiple TCP connections per BoD entity: In this set of simulations we varied the number of connections that are treated per BoD entity. The BoD entity still submits requests to the Controller on the basis of the MAC buffer occupancy but now queues carry the packets from a variable number of TCP connections.

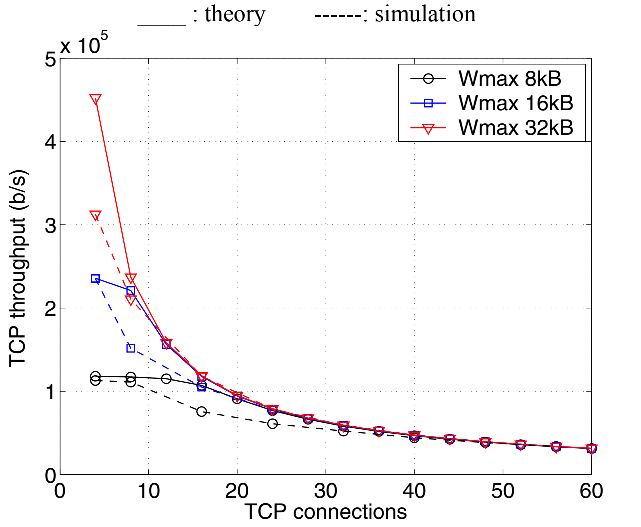

Fig. 2. TCP throughput versus number of connections – one-TCP connection per BoD entity, no link errors

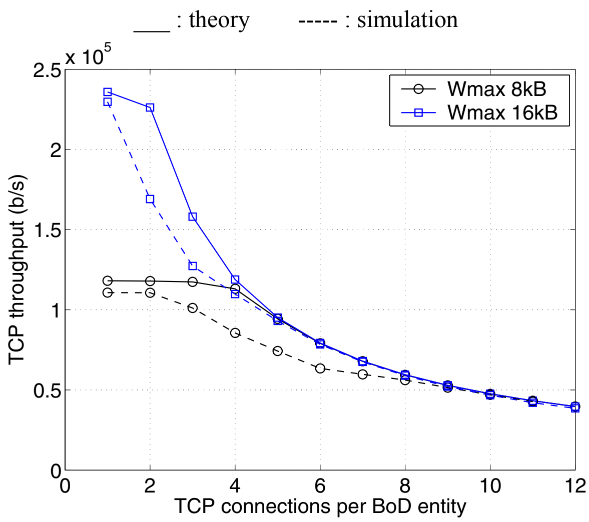

Fig. 3. TCP throughput versus number of connections – four BoD entities, multiple TCP connections per entity, no link errors

The trend is not significantly different than in the single connection per entity scenario (Fig. 3). Again the agreement is closer at higher load and at very low load, where the limiting factor upon the connection throughput is not the bandwidth availability but the maximum window constraint. Apparently, at moderate load levels and for the plotted levels of aggregation the TCP burstiness remains significant so that the TCP packet arrivals are distinctively different than Poisson arrivals.

3) Link errors: The simulation results are compared against the analytical estimates relying on both TCP throughput formulas, the *square root p* and the *PFTK*. The agreement is closer in the second case (Fig. 4), since the connections were subject to rather high error rate –MAC frame error rate (FER) equal to 10^{-3}– where timeouts are non-negligible events.

The analytical method captures the impact of the W_{max} constraint but overall the deviation from the simulation results is higher when compared with the error-free links scenario. The reason is that now the link is saturated for a much higher number of connections than in the error-free links case, since connections are hard-limited by the link errors rather than the bandwidth availability almost throughout the offered load range, in terms of numbers of TCP connections sharing the link.

B. *Finite duration flows*

Regarding simulations, TCP flows arrive according to a Poisson process, whereas the connection duration is assumed exponential and is varied to give rise to different traffic load over the satellite link. The comparisons with the analysis were performed via a Monte Carlo technique, similar to the one described in [8]: for each value of the mean number of connections sharing the satellite link, we perform 500 computations, each one with a different sample N_s. The distribution of N_s should resemble the actual distribution of active connection over the satellite link. We tested two cases for this distribution: the Poisson distribution, corresponding to the M/G/∞ model for connection arrivals [18] and the geometric distribution.

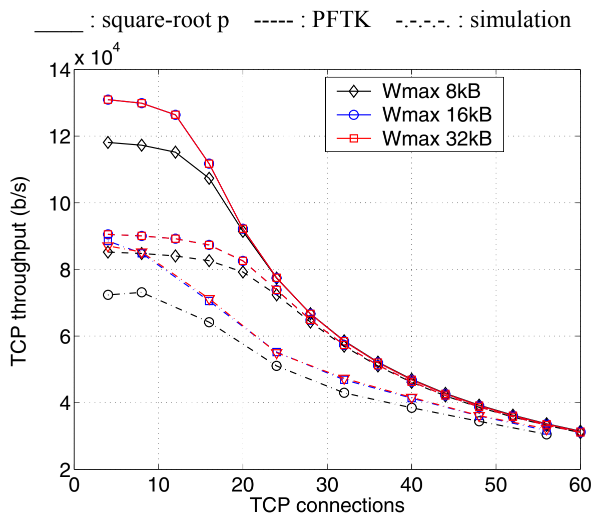

Fig. 4. TCP throughput versus number of connections – one-TCP connection per BoD entity, FER=10^{-3}

Table I and Table II compare analytical estimates with simulation results on the basis of the mean degradation factor F_D. This is estimated over all performed transfers as the ratio

TABLE I
TCP LATENCY FOR MEAN FILE SIZE 50KB

	Degradation factor F_D		
Mean /standard deviation of active connections	Simulation	M/G/∞	Geometric
8.45 / 3.05	1.46	1.01	1.01
20.7 / 6.06	1.91	1.10	1.21
39.5 / 12.14	2.56	1.73	1.96

TABLE II
TCP LATENCY FOR MEAN FILE SIZE 200KB

	Degradation factor F_D		
Mean /standard deviation of active connections	Simulation	M/G/∞	Geometric
9.22 / 3.75	1.56	1.01	1.12
26.1 / 9.72	2.44	1.47	1.72
91.62 / 47.11	5.95	5.19	5.47

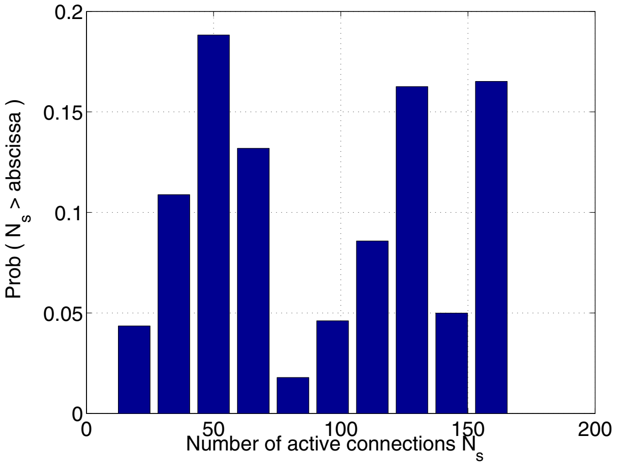

Fig. 5. Probability distribution of the mean active TCP connections over the satellite link (mean file size 200kB –mean active number 91.6)

of the actual time of transfer completion T over the time T_o that would be required under no load (no delay at MAC level)

$$D_F = \frac{T}{T_o} \tag{17}$$

It resembles the "Figure of Demerit" in [1], the only difference being that T_o in (17) is estimated on the basis of the CSA formula for error/loss probability $p - 0$.

Both the Poisson and the Geometric model for the number of active connections underestimate the actual TCP latency, suggesting a different distribution for the TCP flows. In fact, Fig. 5 suggests that the distribution is not even unimodal.

V. SUMMARY-CONCLUSIONS

Our paper describes a fixed-point method for the analytical investigation of the performance of TCP flows in geostationary satellite networks. The method formulation caters for several features of these networks such as asymmetry, broadcast and MAC-shared links, split-TCP connections. The method was applied to the particular case of MAC-shared satellite links with dynamic bandwidth allocation mechanisms, under different scenarios addressing the impact of link errors, traffic aggregation at the satellite gateway nodes and TCP connection type (short or long).

The weakest link in the method chain appears to be the satellite MAC protocol model. Most of the analytical studies in literature in this area assume infinite / finite terminal population producing Poisson packet arrivals (or "sister" distributions– e.g. Bernoulli trials). These models cannot cater for the TCP burstiness. Nevertheless, the analytical estimates of the method are precise at high load ranges and can capture some fundamental dynamics of TCP performance (e.g. W_{max} limitations). Unfortunately, the ultimate limitations to the method applicability have to do with the lack of models addressing the non-homogeneous traffic load case and the traffic prioritization mechanisms at the MAC layer.

REFERENCES

[1] D.P. Heyman, T.V Lakshman and A.L. Neidhardt, "A new method for analyzing feedback-based protocols with applications to engineering web traffic over the Internet", *ACM Sigmetrics performance evaluation review*, vol. 25, no. 1, pp. 24-38, 1997

[2] K. Lindberger, "Balancing quality of service, pricing and utilization in multiservice networks", in *Proc. 16th Int. Teletraffic Congress*, Edinbourgh, Scotland, 1999

[3] A. Berger and Y. Kogan, "Dimensioning bandwidth for elastic traffic in high-speed data networks", *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 643-654, Oct 2000

[4] M. Mathis, J. Semke, J. Mahdavi and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm", *Computer Communications Review*, vol. 3, July 1999

[5] J Padhye, V. Firoiu, D. Towsley and J. Kurose, "Modeling TCP throughput: a simple model and its empirical validation", in *Proc. ACM SIGCOMM 1998,* Vancouver, 1998

[6] N. Cardwell, S. Savage and T. Anderson, "Modeling TCP latency", IEEE INFOCOM, March 2000

[7] M. Roughan, A. Erramilli and D. Veitch, "Network Performance for TCP Networks. Part I: Persistent Sources", in *Proc. 17th Int. Teletraffic Congress*, Salvador, Brasil, 2001

[8] R.J. Gibbens *et al.*, "Fixed-point models for the end-to-end performance analysis of IP networks", in *Proc. 13th ITC Specialist Seminar :IP Traffic Measurement, Modeling and Management*, 2000

[9] C. Casetti and M. Meo, "An analytical framework for the performance evaluation of TCP Reno connections", *Computer Networks*, vol. 37, pp. 669-682, 2001

[10] A. Misra, T. Ott and J. Baras, "Predicting bottleneck bandwidth sharing by generalized TCP flows", *Computer Netwo*rks, vol. 40, pp. 557-576, Oct 2002

[11] T. Bu and D. Towsley, "Fixed Point *approximation for TCP behavior in AQM network",* in *Proc. 2001 ACM Sigmetrics Int. Conf. on Meas. and Model. of Comp. Systems,* Cambridge-Massachusetts, 2001

[12] M. Karaliopoulos and B. Evans, "Modeling split-TCP latency and buffering requirements in GEO satellite networks", unpublished

[13] H. Peyravi, "Medium access control performance in satellite communications", *IEEE Communications Magazine*, pp. 62-71, 1999

[14] T. Le-Ngoc and S. Krishnamurthy, "Performance of combined free/demand assignment multiple-access schemes in satellite communications", *International Journal of Satellite Communications*, vol.14, pp. 11-21, 1996

[15] T.V Lakshman and U. Madhow, "The performance of TCP/IP for networks with high bandwidth-delay products and random loss", *IEEE/ACM Transactions on Networking*, vol. 5, no.3, pp. 336-350, 1997

[16] M. Karaliopoulos, R. Tafazolli and B. Evans, "An ns-derived GEO satellite network simulator: features, capabilities, results", in *Proc. IEE seminar on sim. and modeling of satellite systems*, London, 2002

[17] M. Karaliopoulos, R. Tafazolli and B. Evans, "Providing differentiated service to TCP flows over geostationary BoD satellite networks"', *IEEE Journal of Selected Areas in Communication*, vol. 22, no. 2, pp. 333-347, Feb. 2004

[18] V. Paxson and S. Floyd, "Wide-area traffic: the failure of Poisson modeling", *IEEE/ACM Transactions on Networking*, vol. 3, pp. 226-244, 1996