REFWA: An Efficient and Fair Congestion Control Scheme for LEO Satellite Networks

Tarik Taleb, *Member, IEEE*, Nei Kato, *Senior Member, IEEE*, and Yoshiaki Nemoto, *Senior Member, IEEE*

***Abstract*—This paper examines some issues that affect the efficiency and fairness of the Transmission Control Protocol (TCP), the backbone of Internet protocol communication, in multi-hops satellite network systems. It proposes a scheme that allows satellite systems to automatically adapt to any change in the number of active TCP flows due to handover occurrence, the free buffer size, and the bandwidth–delay product of the network.**

The proposed scheme has two major design goals: increasing the system efficiency, and improving its fairness. The system efficiency is controlled by matching the aggregate traffic rate to the sum of the link capacity and total buffer size. On the other hand, the system min-max fairness is achieved by allocating bandwidth among individual flows in proportion with their RTTs. The proposed scheme is dubbed *Recursive, Explicit, and Fair Window Adjustment (REFWA)*.

Simulation results elucidate that the REFWA scheme substantially improves the system fairness, reduces the number of packet drops, and makes better utilization of the bottleneck link. The results demonstrate also that the proposed scheme works properly in more complicated environments where connections traverse multiple bottlenecks and the available bandwidth may change over data transmission time.

***Index Terms*—Congestion control, fairness, receiver's advertised window adjustment, satellite networks, TCP.**

I. Introduction

THE need for satellite communication systems has grown rapidly during the last few years. Inter-networking with satellites began with the use of individual satellites in geostationary orbits. However, requirements for lower propagation delays and propagation loss, in conjunction with the coverage of high latitude regions for personal communication services, have sparked the development of new satellite communication systems called Low Earth Orbit (LEO) satellite systems [1]. Due to the universality of the Transmission Control Protocol (TCP), an in-depth understanding of TCP and recognition of its merits and drawbacks in LEO satellite networks are of vital importance. This understanding underpins the research work outlined in this paper.

TCP usually results in drastically unfair bandwidth allocations when multiple connections share a bottleneck link [3], [4]. This unfairness issue appears when the number of flows varies over time, as has been indicated in studies of terrestrial wide-area network traffic patterns [2]. Yang *et al.* [5] demonstrate further the unfairness of TCP by addressing its transient behaviors and comparing them to that of some notable TCP-friendly congestion control protocols. In the case of multi-hops satellite constellations, since TCP's throughput is inversely proportional to the Round Trip Time (RTT), the unfairness issue becomes more substantial. Indeed, when a group of users, with considerably different RTTs, compete for the same bottleneck capacity, TCP's undesirable bias against long RTT flows becomes more significant, as is reported in [6]. Additionally, satellite networks are well characterized by frequent handover occurrences [7]. A handover occurrence may force a TCP sender to either suddenly alter its path and compete for bandwidth with a different group of connections, or just keep its path but share the same link with newly incoming connections. Both cases will eventually result in an abrupt change in both the flows count and the connection's RTT. If all TCP senders keep sending data without any adjustment to their sending rates, an increase in the flows count will result in overloading the link with data packets causing excessive queueing delays, large number of packet drops, and throughput degradation, whereas a decrease in the flows count will lead to a waste of bandwidth and ultimately lower link utilization.

As a remedy to the above issues, this paper argues that the TCP rate of each flow should be adaptively adjusted to the available bandwidth when the number of flows that are competing for a single link, changes over time. An explicit and fair scheme is developed. The key concept of the proposed scheme is to match the aggregate window size of all active TCP flows to the effective network bandwidth–delay product. At the same time, the scheme provides all the active connections with feedbacks proportional to their RTT values so that the system converges to optimal efficiency and max-min fairness. Feedbacks are signaled to TCP sources through the receiver's advertised window (RWND) field in the TCP header of acknowledgments. The proposed scheme is dubbed *Recursive, Explicit, and Fair Window Adjustment (REFWA)*.

Simulation results reveal that the REFWA scheme substantially improves the system fairness, reduces the number of packet drops, and makes better utilization of the bottleneck link. The results also demonstrate that the proposed scheme works properly in more complicated environments where there are multiple bottlenecks, and the available bandwidth is not always steady, but may change over data transmission time. An abridged version of this paper can be found in [8].

The remainder of this paper is structured as follows. Section II highlights the relevance of this work to the state-of-the-art in the context of TCP performance over satellite networks. The key design philosophy and distinct features that were incorporated in the proposed scheme are described in Section III. Section IV

Manuscript received August 17, 2004; revised April 18, 2005, and August 13, 2005. This work was supported by the Koden Electronics Company, Ltd.

The authors are with the Graduate School of Information Sciences, Tohoku University, Sendai Miyagi 980-8579, Japan (e-mail: taleb@nemoto.ecei.tohoku.ac.jp; kato@nemoto.ecei.tohoku.ac.jp; nemoto@nemoto.ecei.tohoku.ac.jp).

Digital Object Identifier 10.1109/TNET.2006.883130

portrays the simulation philosophy in addition to defining various details for the setting of parameters. Simulation results are presented in Section V. Section VI discusses some implementation issues and previews some of the extension work that is under investigation to further improve the overall performance of the proposed scheme. The paper concludes in Section VII with a summary recapping the main advantages and achievements of the proposed scheme.

II. State-of-the-Art Related Work

After nearly two decades since its standardization, the TCP protocol has seen deployment at unforseen scale. Despite the widespread acceptance of TCP/IP in terrestrial networks, its performance over satellite networks is still limited due to a number of reasons related to the protocol syntax and semantics [9]. The remainder of this section describes the main post-standard improvements that have been devised in recent literature to overcome the shortcomings of TCP in satellite networks.

To diminish the negative impact slow start has on performance, the congestion window is initially set to a value larger than one packet but smaller than four packets in size [10]. Despite the advantages of this operation in improving the TCP throughput, there are several problems. Most notably an increased value of the initial window would increase the burstiness of the sender and could exacerbate existing congestion in a network. Akyildiz *et al.* [11] have investigated the usage of low-priority dummy segments to probe the availability of network resources without carrying any new information to the sender. In TCP/SPAND [12], network congestion information is cached at a gateway and shared among many co-located hosts. Using this congestion information, TCP senders can make an estimate of the optimal initial congestion window size at both connection start up and restart after an idle time. Other researchers have discussed the potential of a technique called TCP Spoofing [13], [14]. In this technique, a router near the TCP sender prematurely acknowledges TCP segments destined for the satellite host. This operation gives the source the illusion of a short delay path speeding up the sender's data transmission. Another similar concept is TCP Splitting where a TCP connection is split into multiple connections with shorter propagation delays [15].

In short, many researchers have investigated the performance of TCP in satellite networks. However, the central theme in their pioneering studies pertains only to efficiency issues, mainly to problems related to the slow-start phase, whereas TCP behavior under many competing flows in satellite networks has not been sufficiently explored. It is worth noting that all of the improvements described so far are confined to the TCP entities at the connections end-points. The rest of this section surveys the recently proposed mechanisms that require network infrastructure support, e.g., changes in Internet routers.

Current TCP implementations do not communicate directly with the network elements for explicit signaling of congestion control. TCP sources infer the congestion state of the network only from implicit signals such as the arrival of ACKs, timeouts, and receipt of duplicate acknowledgments (dupACKs). In the absence of such signals, the TCP congestion window increases to the maximum socket buffer advertised by the receiver. In the case of multiple flows competing for the capacity of a given link, this additive increase policy will cause severe congestion, degraded throughput, and unfairness.

One approach to control congestion is to employ scheduling mechanisms, fair queueing, and intelligent packet-discard policies such as Random Early Marking (REM) [16] and Random Early Discard (RED) [17] combined with Explicit Congestion Notification (ECN) [18]. These policies require a packet loss to create an early signal of the network congestion to TCP sources. However, in the case of large delay links (e.g., satellite links), these policies become inefficient and may still cause timeouts forcing TCP senders to invoke the slow-start phase. By the time the source starts decreasing its sending rate because of a packet loss, the network may already be overly congested. Low *et al.* [19] have shown through mathematical analysis the inefficiency of these Active Queue Management schemes (AQM) in environments with high bandwidth–delay product such as satellite networks.

AQM limitations can be mitigated by adding some new mechanisms to the routers to complement the endpoint congestion avoidance policy. These mechanisms should allow network elements between a TCP source and a TCP destination to acknowledge the source with its optimal sending rate. By so doing, the whole system becomes self-adaptive to traffic demands and more active in controlling congestion and buffer overflows. To cope with TCP limitations in high bandwidth–delay product networks, several studies have been conducted providing valuable insight into TCP dynamics in such environments. TCP-Vegas [20] attempts to compute the optimal setting of the window size based on an estimate of the bandwidth–delay product for each TCP connection. As knowledge of the RTT and the bandwidth–delay product of the network is not usually available at network elements, TCP-Vegas requires extensive modifications to current TCP implementations in end-systems.

Katabi *et al.* [21] proposed a new congestion control scheme, eXplicit Control Protocol (XCP). The scheme substantially outperforms TCP in terms of efficiency in high bandwidth–delay product environments. However, the main drawback of this protocol is that it assumes a pure XCP network and requires significant modifications at the end-system. Explicit Window Adaptation (EWA) [22] and WINdow TRAcking and Computation (WINTRAC) [23] suggest an explicit congestion control scheme of the window size of TCP connections as a function of the free buffer value similar in spirit to the idea of Choudhury *et al.* [24]. Since the computed feedback is a function of only buffer occupancy and does not take into account link delay or link bandwidth, the two schemes are likely to run into difficulty when faced with high bandwidth or large delay links similarly to AQMs. Additionally, EWA and WINTRAC achieve fairness only in the distribution of the maximum achievable window size. The two schemes remain grossly unfair towards connections with high variance in their RTTs distribution.

In the sphere of TCP behavior under many competing flows in satellite networks, [25] and [26] appear to be the first comprehensive papers that introduce the idea of explicit signaling and jointly address both efficiency and fairness of TCP in broadband satellite networks. The basic idea behind the two papers is to adjust the window size of all active TCP flows to the network

pipe which was assumed to be constant during the data transmission time. The papers considered only a simple network environment. It was assumed that only one bottleneck was traversed by many connections and that all flows always made full use of their allocated windows. It was also assumed that the bandwidth available to TCP flows remained steady over time. However, these assumptions are not generally valid and different situations are likely to pose some limitations on the performance of the proposed solution. For instance, when there are multiple bottlenecks, a TCP connection may send data with rates fed back by a given bottleneck but smaller than the windows allocated by the other bottlenecks. This will obviously cause the underutilization of the latter. In addition, due to the variability of higher priority traffic, the available bandwidth is not always steady and may change over data transmission time. These kind of situations may attenuate the performance of the scheme proposed in [25] and [26].

III. Operational Overview of the REFWA Scheme

This section presents an operational overview of the feedback computation. Before delving into details of the proposed scheme, it should be stated that this research work considers the type of satellite constellations where satellites are supported by routing and path pinning capabilities at flow level. The Inter-Satellite Link (ISL) delay is assumed to be constant as well. First, there is a description of how the scheme exploits some features of satellite networks to make an approximate estimate of flows RTTs and the bandwidth–delay product of the network.

A. Key Concepts of REFWA

1) Estimate of Flows RTT: Prior knowledge of the RTT estimates is usually not available at network elements in terrestrial networks. However, in the case of multi-hops LEO satellite constellations, flows RTT can be handled by a simple monitoring of hops count in the backward and forward traffic of each flow. The hops count of each flow can be easily computed from the time-to-live (TTL) field in the IP header of both ACK and data packets [27]. Let H_b and H_f denote the number of hops traversed by an acknowledgment packet in the backward traffic and the number of hops traversed by a data packet in the forward traffic before entering the router in question, respectively. Having the ISL delay constant and omitting the contribution of queueing delays in the one-way propagation delay, the connection RTT estimate can be approximated as

$$\mathrm{RTT} \approx 2 \cdot (H_b + H_f + 2) \cdot \mathrm{ISL}_{\mathrm{delay}} \tag{1}$$

where $\mathrm{ISL}_{\mathrm{delay}}$ denotes the ISL delay. Admittedly, it should be emphasized that with almost full utilization of network resources, queueing delays can not be negligible and may represent a significant portion of the total propagation delay. However, the interest of this paper is not to have an accurate estimate of the connections RTTs but rather approximate values of RTTs. Indeed, as it will be clarified by the simulation results presented in the paper, the omission of queueing delays in the RTT estimate does not hinder the good performance of the proposed scheme, and that is the case even in the presence of bottlenecks.

2) Flows State Table: Using knowledge of the RTT estimates, flows are grouped according to the number of hops they traverse. Each group of flows is identified by a group ID. At each satellite, a group G is defined as the set of flows that traverse the same number of hops and thus have equal RTTs, RTT_G. Flows are identified by a flow ID and are defined as streams of packets sharing the quintuple: source and destination addresses, source and destination port numbers, and the protocol field. In the case of IPv6, the flow ID can be simply deduced from the flow label. A flow is considered to be in progress if the time elapsed since its last packet transmission time is inferior to a predetermined threshold δ. This threshold δ is periodically updated to the most recent estimate of the average RTT of active flows, $\mathrm{RTT}_{\mathrm{avg}}$ ($\delta = \mathrm{RTT}_{\mathrm{avg}}$).

To best explain the computation of the threshold δ, let's consider two time slots, $[t_{n-1}; t_n]$ and $[t_n; t_{n+1}]$. Let us assume that a satellite router is currently performing within the time slot $[t_n; t_{n+1}]$. At the current time, the satellite router knows about the number of active flows that traversed the satellite during the previous time slot $[t_{n-1}; t_n]$, and the average of their RTT estimates,[1] $\mathrm{RTT}_{\mathrm{avg}}$. The threshold δ of the current time slot $[t_n; t_{n+1}]$ is simply set to the $\mathrm{RTT}_{\mathrm{avg}}$ of the previous time slot. And the time slot $[t_n; t_{n+1}]$ is decided in such a way that its length is equal to $\mathrm{RTT}_{\mathrm{avg}}$ ($\mathrm{RTT}_{\mathrm{avg}} = t_{n+1} - t_n$). Flows are counted periodically every $\mathrm{RTT}_{\mathrm{avg}}$ time and routers on-board the satellites should create a new entry in the table whenever a new flow is detected. Note that estimating parameters over intervals longer than $\mathrm{RTT}_{\mathrm{avg}}$ will tend to ignore short flows leading to a sluggish response, while estimating parameters over shorter intervals will lead to erroneous estimates which ultimately affect the effectiveness of the proposed scheme [28], [29].

3) Correlation Between a Connection RTT and Its Throughput: Due to the fundamental dynamics of TCP, there is a significant bias towards shorter connections when considering the individual throughput of connections with different RTTs. Being interested in quantifying this discrimination, several studies have shown that the average throughput of a TCP connection is inversely proportional to RTT^{β} [4], [30], where *RTT* is the connection's round-trip time and $1 \leq \beta < 2$. In order to make the system converge to max-min fairness, the proposed scheme exploits this attribute as will be shown hereafter.

B. Feedback Computation Method

Let $\mathrm{RTT}_{\mathrm{avg}}$ denote the average RTT of all active flows traversing a satellite. At time ($t = n \cdot \mathrm{RTT}_{\mathrm{avg}}$), the feedback value of flows that belong to the κth group is computed as

$$\begin{aligned} F_\kappa(n) &= \frac{\mathrm{RTT}_\kappa^\alpha}{\sum_{j=1}^{M} n_j \cdot \mathrm{RTT}_j^\alpha} \cdot \Upsilon(n) \\ \Upsilon(n) &= \Upsilon(n-1) + \phi\Big(Bw \cdot \mathrm{RTT}_{\mathrm{avg}} - \Upsilon(n-1)\Big) \\ &\quad + \psi\Big(B - Q(n-1)\Big) \end{aligned} \tag{2}$$

where M, B, and Bw are the total number of flows groups, the router's buffer size, and the link bandwidth, respectively.

[1]This value is referred to as the most recent estimate of the average RTT, and can be counted using the Exponentially Weighted Moving Average (EWMA) algorithm.

n_j and RTT_j denote the size of the jth group and the RTT value of its flows, respectively. $\Upsilon(n)$ and $Q(n)$ denote the aggregate TCP window size and the router's queue occupancy at time $(t = n \cdot \mathrm{RTT}_{\mathrm{avg}})$. α, ϕ, and ψ are constant parameters, whose influence on the system efficiency and fairness will be discussed later in the simulation results. In the rest of this paper, α is referred to as the skew factor of the system. Notice that setting α to values smaller than one aggravates TCP's undesirable bias towards short RTT flows while setting α to values larger than one yields a bias in favor of long RTT flows. Observe also that ϕ and ψ play a significant role in exploiting the system spare bandwidth and free buffer size, respectively. In other words, in the case of large values of ϕ and ψ, when some connections are not making full use of their allocated bandwidths during the interval time $[(n-1)\mathrm{RTT}_{\mathrm{avg}}, n\mathrm{RTT}_{\mathrm{avg}}]$, the spare bandwidth, $(Bw \cdot \mathrm{RTT}_{\mathrm{avg}} - \Upsilon(n-1))$, will increase and the buffer occupancy, $Q(n-1)$, will go down causing an increase in the computed feedbacks of the other connections at time $(t = n \cdot \mathrm{RTT}_{\mathrm{avg}})$. This will help to fully utilize the link capacity while maintaining small buffer sizes. It should be recognized also that $\Upsilon(n)$ is computed in a recursive manner, hence the name of the proposed scheme, *Recursive, Explicit, and Fair Window Adjustment (REFWA)*

One of the most interesting attributes of this feedback computation method is that it allows the system to automatically adapt to the number of active TCP flows,[2] the free buffer size,[3] and the bandwidth–delay product of the network. As previously discussed, most traditional self-adaptive schemes base their computed feedback on only buffer occupancy and do not take into account link delay or link bandwidth. Similarly to AQMs, these schemes are inefficient in environments with high bandwidth–delay product such as satellite networks [19]. The addition of the bandwidth–delay product of the network to the feedback computation aims to deal with such an issue. Another benefit of the proposed feedback computation method is that it controls the efficiency of the system by matching the aggregate traffic rate to the link capacity and total buffer size. This attribute eventually helps to adjust the protocol's aggressiveness and to prevent persistent queues from forming. To achieve min-max fairness, the first term of (2) re-allocates bandwidth between individual flows in proportion with their RTTs.

C. Feedback Signaling Method

The window feedback is computed every $\mathrm{RTT}_{\mathrm{avg}}$ time and is written in the receiver's advertised window (RWND) field carried by the TCP header of acknowledgment packets similarly in spirit to the EWA [22] and WINTRAC [23] approaches. RWND adjustment requires the ability of the router to separate ACK packets from data packets. This ACK packet identification can be easily carried out by examining the ACK bit in the TCP packet header. Indeed, ACK packets have always their ACK bit set by the TCP receiver. On the other hand, RWND adjustment does not require modifications to the protocol implementations in the end system, nor does it modify the TCP protocol itself.

[2]Similarly in spirit to the idea of [28].

[3]Similarly to self-adaptive mechanisms (e.g., EWA [22] and WINTRAC [23]).

The RWND value can only be downgraded. If the original value of the receiver's advertised window, which is set by the TCP receiver, exceeds the feedback value computed by a downstream node, the receiver's advertised window is reduced then to the computed feedback value. A more congested router later in the path can further mark down the feedback by overwriting the receiver's advertised window field. Ultimately, the RWND field will contain the optimal feedback from the bottleneck along the path. Values of RWND smaller than the optimal value decrease the throughput and result in link under-utilization, whereas larger values of RWND contribute only to increased queueing delays, multiple drops, and hence throughput degradation. When the feedback reaches the sender, the sender reacts to the message and accordingly updates its current window. This dynamic adjustment of RWND value will help to smooth the TCP burstiness and to achieve efficiency and a fair window size distribution among all competing flows.

IV. Simulation Set-Up

This section gives a detailed description of the simulation environment, justifying the choices made along the way. The design of the simulation setup relies on Network Simulator (NS) [31]. Particular attention is thus paid to the design of an accurate and realistic one. Unless otherwise noted, the parameters specified below are those used in all the experiments throughout the paper.

In all simulations, TCP sources implement the TCP NewReno version [32]. TCP NewReno achieves faster recovery and has the potential of significantly improving TCP's performance in the case of bursty losses. Whilst it has been observed that many TCP connections in today's Internet traffic are of short durations, the behavior of the proposed scheme is best understood by considering persistent sources, which could be thought of as modeling long file transfers. TCP connections are, therefore, modeled as greedy long-lived FTP flows. These long-lived FTP flows can serve to model real video streams as well, where efficiency and fairness issues matter as the number of video-data receivers increases. The data packet size is fixed to 1 kB. In order to remove limitations due to small buffer sizes on the network congestion, buffers equal to the bandwidth–delay product of the bottleneck link are used [33]. Throughput and queue size measurements are performed in intervals of 100 ms. Due mostly to its simplicity and its wide usage in today's switches and routers, all satellites use Drop-Tail as their packet-discarding policy. Simulations were all run for 20 s, a duration long enough to ensure that the system has reached a consistent behavior. In the conducted simulations, the real satellite motion is not included. Indeed, the satellite constellation is considered dynamic through the concept of dynamic virtual topology. In this way, the network is modeled as a set of time-discrete snapshots of satellite positions over one system cycle T, which can be divided into a number of time intervals with variable lengths $([t_0 = 0; t_1], [t_2; t_3], \ldots, [t_{m-1}; t_m = T])$. Over each interval, the topology is considered to be constant; the link state changes take place at only discrete times $(t_0, t_1, \ldots, t_m)$. Having the simulation running time set to 20 s, a value significantly smaller than the interval lengths, the satellite motion can be neglected during the entire running time of the simulations.

As for the links capacity, all up-links and down-links are given a capacity equal to 10 Mb/s. The ISL delay, $\mathrm{ISL}_{\mathrm{delay}}$, is set to 20 ms. These parameters are chosen with no specific purpose in mind and do not change any of the fundamental observations about the simulation results. Unless otherwise stated, all links are presumed to be error-free throughout this paper. This assumption is made so as to avoid any possible confusion between throughput degradation due to packet drops and that due to satellite channel errors.

To illustrate the issues at hand, a satellite network is modeled as a single network bottleneck as depicted in Fig. 1(a) and (b). The bottleneck link is composed of three satellites. For the sake of simplicity, senders/receivers are assumed to traverse different set of satellites before entering/leaving the bottleneck, as shown in the two figures. This assumption has no effect on the overall performance of the proposed method. To evaluate the performance of the proposed scheme in case of multiple-bottlenecks environments, Fig. 1(c) considers the case of two bottlenecks being shared by three groups of flows.

The remainder of this section presents the quantifying parameters that are used to evaluate the performance of the proposed scheme. In addition to TCP sequence number and individual flow throughput metrics that give good insight into the behavior of individual flows, the following measures will be used to capture the aggregate behavior of flows and the overall network performance:

- Bottleneck link utilization: The ratio of the aggregate throughput to the bottleneck link capacity. This measure involves the aggregate traffic behavior and indicates the efficiency of the protocol.
- Loss rate: The ratio of the dropped packets to the aggregate sent packets.
- Queue size: The queue size of the router measured every 100 ms.
- Average queue size: The average queue occupancy at the router averaged over the transmission time.
- Fairness index: This measure is defined in [34]. The fairness index involves the relative throughput of flows sharing a link defined as

$$F(x) = \frac{\left(\sum_{i=1}^{N} \frac{x_i}{b_i}\right)^2}{N \cdot \sum_{i=1}^{N} \left(\frac{x_i}{b_i}\right)^2}$$

where x_i is the actual throughput of the ith flow and b_i is the equal share of the bottleneck link capacity. The fairness index of a system ranges from zero to one. Low values of the fairness index represent poor fairness among the competing flows. Depending on the application and the number of TCP senders, gaining higher fairness values is sometimes worthwhile even at the cost of reduced efficiency.

V. Performance Evaluation

Having described the simulation parameters, we now direct our focus to evaluating the performance of the REFWA scheme through extensive simulation experiments. In the performance evaluation, TCP NewReno is used as a comparison term and comparisons are made in both steady-state and dynamic environments.

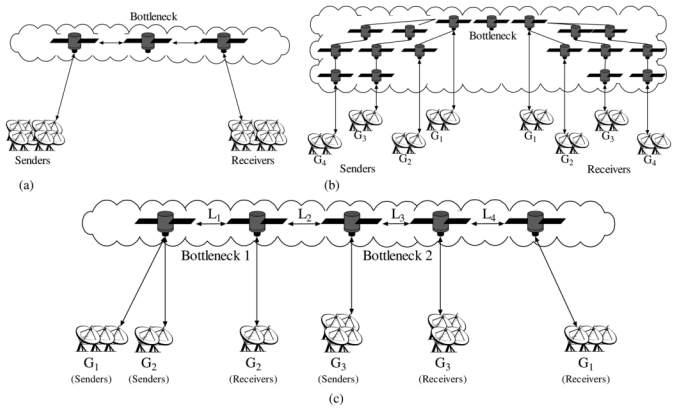


Fig. 1. Simulation environments. (a) Single bottleneck shared by flows with equal RTTs. (b) Single bottleneck shared by flows with different RTTs. (c) Network with multiple bottlenecks.

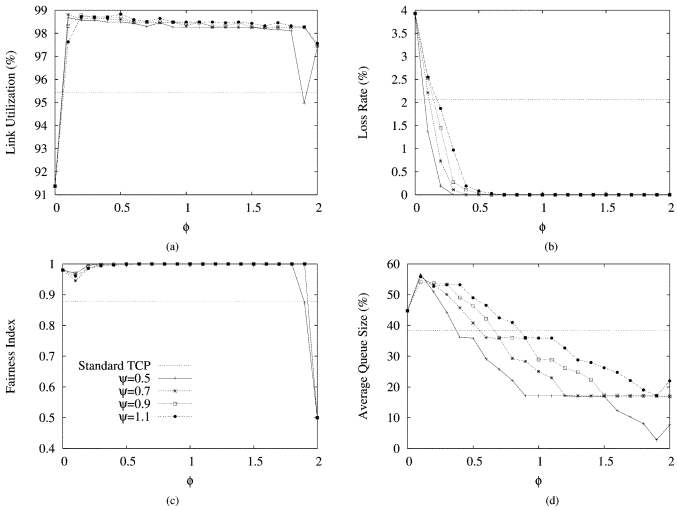


Fig. 2. Effect of ϕ and ψ on the overall network performance. (a) Link utilization (%). (b) Loss rate (%). (c) Fairness index. (d) Average queue size (%).

A. Effect of ϕ and ψ

In this experiment, the used network configuration is that of Fig. 1(a). The ISL bandwidth is set to 1.544 Mb/s (e.g., T1). Eight connections are activated at the beginning of the simulation. Their starting times are uniformly distributed from 0 to 5 ms to avoid bursty losses at the simulation launch time. All connections remain open for 20 s, the simulation running time. Fig. 2 graphs the overall network performance in terms of link utilization, drops rate, fairness, and average queue size for different values of ϕ and ψ. Simulation results show that large values of ψ cause higher packet losses and larger queue sizes. This can be explained by the fact that ψ aims to control the free buffer size: large values of ψ cause large bursts in the network which obviously lead to larger queue sizes and, ultimately, a higher number of packet drops. The simulation results indicate also that the system experiences a significantly higher number of drops and larger queue size in case of small values of ϕ. It is observed also that the overall throughput gets

degraded when ϕ takes values in the vicinity of zero, whereas the system fairness remains unaffected. The main reason behind this performance is that in case of small values of ϕ, the effect of ϕ in the feedback computation becomes minimal and the proposed scheme becomes, in nature, similar to traditional self-adaptive schemes where the feedback computation is based on only the buffer occupancy (e.g., EWA [22] and WINTRAC [23]). In deed, having similar RTTs, flows are assigned similar portions of the bandwidth. This explains the higher value of the fairness index. However, as the bandwidth–delay product of the network is not used in the feedback computation, the allocated portions are smaller than the bandwidth flows can actually be using for sending data. This explains the lower throughput. This throughput degradation demonstrates the limitation of traditional self-adaptive schemes in environments with high bandwidth–delay products, such as satellite networks. On the other hand, it is observed that large values of ϕ reduce the average queue size, yet degrade the system throughput and considerably affect its fairness.

Being interested in setting ϕ and ψ to fixed values so no further tuning of the system is required, we have developed a mathematical model in Appendix I. The model helps to find the optimal values of ϕ and ψ that guarantee the system's stability. In the remainder of this paper, unless otherwise specified, ϕ and ψ are set to 1.5 and 0.5, respectively.

B. Robustness to Change in Traffic Dynamics

This experiment aims to illustrate the effect of the change in flows count on TCP behavior and to examine how the REFWA scheme adapts to sudden increases or decreases in traffic demands. Similarly to the previous experiment, a symmetric and simple network bottleneck shared by eight connections is considered [Fig. 1(a)]. As for the change in traffic demands, the following scenario is considered. The simulation is launched with four flows (first to fourth) that are let active for 5 s. At time $t = 5$ s, we start another four flows (fifth to eighth) and let them stabilize. At time $t = 15$ s, the last four flows are deactivated. The remaining connections are left active until the end of the simulation. Due to handover occurrence in a multi-hops LEO satellite network, a TCP connection may either alter its path and compete for bandwidth with a different group of connections, or just keep its path but be forced to share the same link with newly incoming connections. Both cases will result in an abrupt change in the number of flows. This justifies the choice of the stair-step form as the form of changes in traffic dynamics in the considered scenario.

In order to demonstrate how the REFWA scheme achieves better fairness, the growth of TCP sequence numbers for each simulated TCP connection is plotted (Fig. 3). With REFWA, the slope of the sequence number lines decreases at time $t = 5$ s as four new flows enter the system, and increases at time $t = 15$ s as a result of extra bandwidth becoming available for the remaining active connections. Observe how REFWA helps all flows to progress evenly and to behave in an identical way, whereas, in case of only standard TCP, all TCP flows exhibit great deviations. Notice also that the slope of the sequence number growth of the newly entering flows exhibits some small oscillations or remains steady most probably due to timeouts. However, with REFWA,

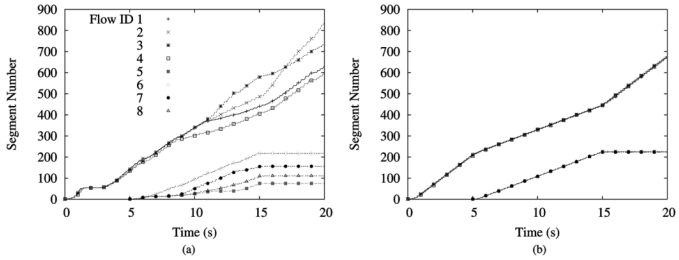


Fig. 3. Sequence number growth of the eight simulated flows ($\phi = 1.5$, $\psi = 0.5$). (a) TCP. (b) REFWA.

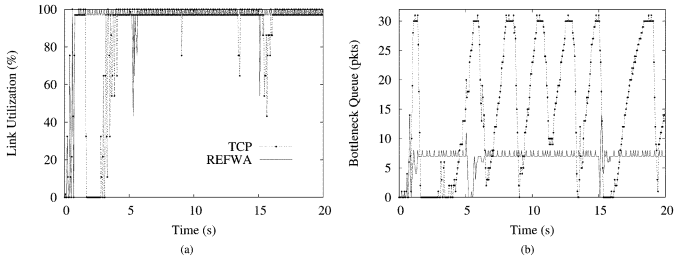


Fig. 4. Bottleneck link utilization and queue size measured every 100 ms time intervals ($\phi = 1.5$, $\psi = 0.5$) (a) Bottleneck link utilization. (b) Bottleneck queue.

it is observed that none of the simulated flows experienced a timeout and that the increase in the sequence number of all flows is parallel. The underlying reason for this performance is in the REFWA's ability to rapidly adapt to changing network conditions and to fairly divide the available bandwidth among the competing flows. When new connections enter the network, the REFWA scheme quickly computes a smaller window feedback, and when some connections exit the network, it feeds back TCP sources with larger window sizes. Additionally, the REFWA scheme does not allow flows to obtain portions of the available bandwidth larger than their fair share values. Consequently, when a flow gets some of its packets dropped, the flow will always be guaranteed a fair portion of the available bandwidth. The flow will then use the allocated portion to recover from losses without entering the slow-start phase that can be triggered if timeouts occur. This assures a fair progression in window size for both newly incoming and already existing flows.

REFWA outperforms the standard TCP not only in terms of fairness and packet drops, but also in improving the link utilization and minimizing the queue size. The queue occupancy of the bottleneck link and the link utilization, measured over intervals of 100 ms, are presented in Fig. 4. The figure indicates that without REFWA, the link utilization fluctuates irregularly. In fact, TCP sources constantly probe for available bandwidth in the network until there is no space in the pipe to maintain new packets. At that time, drops become inevitable. The throughput loss is mainly due to the synchronization of these packet losses and their simultaneous recovery. These synchronized losses cause the connections to simultaneously reduce their windows and ultimately result in degraded throughput. In contrast, REFWA's utilization is always near the total capacity of the link and exhibits limited oscillations; except at time $t = 0$ s and $t = 5$ s due to the entry of many connections all at the same time.

Fig. 4(b) demonstrates that without REFWA, TCP senders increase their window size until they cause buffer overflows. This cycle occurs repeatedly and causes the queue size to oscillate more frequently. Changes in traffic demands result also in transient overshoots in the queue size. These transient overshoots and the large oscillations in the queue size can cause timeouts and frequent underflows, thereby resulting in a substantial idling of the bottleneck link.

On the other hand, in the case of the REFWA scheme, the system is always self-adaptive to the traffic load and the number of active connections. The aggregate value of the window feedbacks of all active TCP flows remains bound to the bandwidth–delay product of the network. This attribute allows the scheme to be effective at protecting the buffer from overflows. With REFWA, the buffer underflows, mostly due to the change in traffic demands are brief and do not significantly affect the bottleneck link utilization. Whilst REFWA aims to reduce queue sizes to a minimum by preventing persistent queues from forming, it is observed that the obtained bottleneck queue size remains constant and is larger than a certain number of packets. This is mainly due to the simultaneous release of entire windows of packets in a single burst at the beginning of each RTT. One possible solution to this issue is the transmission of packets in a steady stream (multiple, small bursts) over the entire course of the RTT [35].

In conclusion, the REFWA scheme manages to control the buffer occupancy well and achieves stability and robustness when a change in traffic load occurs. The average link utilization, as graphed in Fig. 4(a), is perfect most of the time.

C. Fairness

To explore the performance of REFWA in environments with high variance in the RTT distribution, the network topology depicted in Fig. 1(b) is considered. All flows are grouped in four equally-sized groups. Flows belonging to the ith group traverse $(2i + 1)$ satellites causing each flow to have a RTT of $((2i+2)\cdot 40$ ms$)$. A number of test scenarios was created by setting the ISL bandwidth to different typical link speeds: 1.5 Mb/s (e.g., T1), 10 Mb/s (e.g., T2), 45 Mb/s (e.g., T3), and 155 Mb/s (e.g., OC3). For each link type, the maximum number of flows, *MAX*, is fixed so that a minimum value of link fair-share can be guaranteed for all competing flows. In this experiment, the flows count remains constant throughout the simulation and is equal to *MAX*. All connections are activated at the beginning of the simulation and remain open during the simulation running time. Their starting times are uniformly distributed from 0 to 5 ms to avoid bursty losses at the simulation launch time.

First is an investigation of the effect of the skew factor α on the system performance. Fig. 5 presents the overall network performance in terms of link utilization, drops rate, fairness index, and average queue size for different values of α. The figure presents the system performance when the bottleneck link is T1. However, the same experiments were repeated for other link speeds and identical results were obtained. The figure demonstrates that the system fairness decreases significantly when α takes values larger than one. This is because setting α to values larger than one yields a bias in favor of long RTT flows. This bias causes the long RTT flows to send data with rates higher than

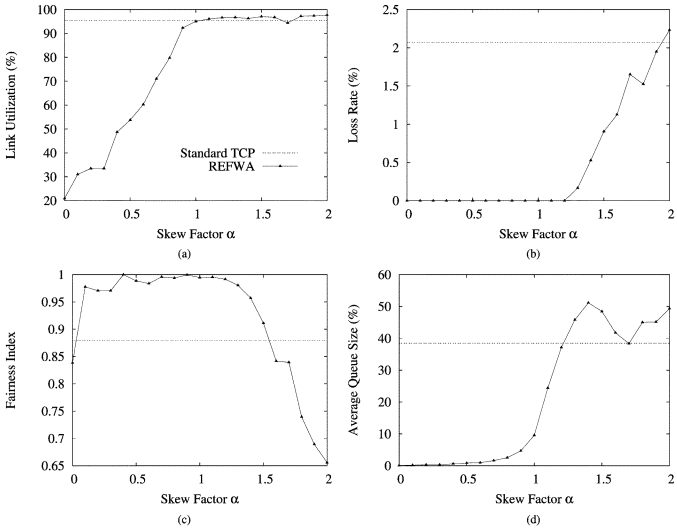


Fig. 5. Effect of α on the overall network performance in case of link type T1 ($\phi = 1.5$, $\psi = 0.5$). (a) Link utilization (%). (b) Loss rate (%). (c) Fairness index. (d) Average queue size (%).

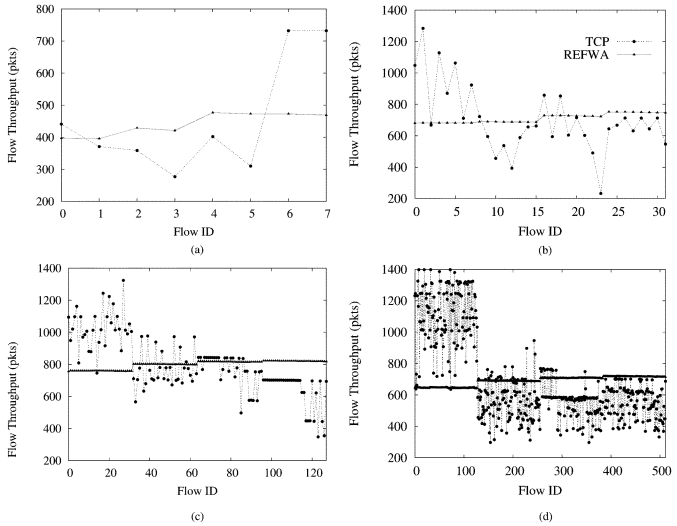


Fig. 6. Individual flow throughput ($\phi = 1.5, \psi = 0.5, \alpha = 1$). (a) Access link T1 ($MAX = 8$ flows). (b) Access link T2 ($MAX = 32$ flows). (c) Access link T3 ($MAX = 128$ flows). (d) Access link OC3 ($MAX = 512$ flows).

their fair shares resulting in traffic bursts. On the other hand, short RTT flows are allocated portions of bandwidth smaller than what it is required to achieve optimal performance. This behavior ultimately causes the system to perform poorly. This poor performance is manifested in the form of lower values of fairness index, higher loss rates, and larger queue sizes. Values of α smaller than one, however, cause the system to converge to optimal fairness (higher values of fairness index) yet result in a significant degradation of the link utilization. It is observed that the system performs well when α is in the vicinity of one. Unless otherwise stated, the skew factor α is set to one in the remainder of this paper.

The individual throughput of the simulated flows is plotted in Fig. 6. Unlike standard TCP, which is grossly unfair towards connections with higher RTTs, the REFWA scheme attempts to fairly divide the available bandwidth among all competing flows

while taking into account the RTT of each flow. This has led to a fair progression in window size for all flows: all the flows could send nearly the same amount of packets in case of the REFWA scheme. The figure demonstrates also the greediness of standard TCP: short RTT connections are seen to conquer most of the link bandwidth and send significantly larger number of packets compared to the long RTT connections. The obtained results confirm as well that the REFWA estimate of the average RTT of the system operates correctly in environments with high variance in the RTT distribution.

D. Resiliency to Errors in RTT Estimate

As described previously, the feedback computation is based on an approximate estimate of flows RTTs. From the conducted simulation results, it was verified that the RTT value estimated from (1) was different than the actual one by 30% to 45% on average. Yet, the simulation results presented so far indicate clearly the better performance of REFWA compared to standard TCP. This section aims to show further that the REFWA scheme maintains a fairly good performance even in the presence of additional RTT estimation errors. In this experiment, eight flows with different RTTs were considered and the configured network is that of Fig. 1(b).

Let RTT_i and $\hat{\mathrm{RTT}}_i$ denote the good estimate and erroneous estimate of RTT of the ith flow. e_i denotes the RTT estimation error ratio of the ith flow and is defined as follows:

$$e_i = \frac{\hat{\mathrm{RTT}}_i - \mathrm{RTT}_i}{\mathrm{RTT}_i}.$$

It should be noticed that the RTT estimation is made based on the number of hops. Therefore, any RTT estimation error value[4] should be an even multiplication of $\mathrm{ISL}_{\mathrm{delay}}$. In this simulation, in order to study the overall performance of the system for different RTT error values, this fact is, however, ignored, and error ratios are deliberately derived from a uniform distribution with a zero mean and a variance of σ^2. In NS implementation, the extreme case is considered by setting the maximum and minimum values of the distribution to σ and $-\sigma$, respectively ($\max\_ = \sigma$, $\min\_ = -\sigma$).

Fig. 7 presents the overall network performance in terms of link utilization, drops rate, and fairness index for different values of σ. The figure demonstrates that for reasonable values of σ, the REFWA scheme still maintains good performance: better link utilization and higher values of fairness index. However, large values of σ worsen the system performance. This aggravation is manifested in the form of link utilization degradation. From these results and given the fact that the RTT estimate was different than the actual one by 30% to 45% on average, it can be concluded that the proposed scheme shows sufficient tolerance to RTT estimation errors.

E. Web-like Traffic

Since a large number of flows in today's Internet are short WEB-like flows, the interaction and resulting impact of such dynamic traffic on the REFWA scheme are discussed in this section. It has been reported in [36] that WEB-like traffic tends to be self-similar in nature. In [37], it is shown that self-similar traffic can be modeled as several ON/OFF TCP sources whose ON/OFF periods are drawn from heavy-tailed distributions such as the Pareto distribution. In this experiment, a scenario where a mix of ten long-lived FTP flows and a number of non-persistent flows compete for the bottleneck link bandwidth is considered. The considered network configuration is that of Fig. 1(a) and the used access link type is T2. The simulation starts with ten persistent connections at time $t = 0$ s. The persistent TCP flows remain open until the end of the simulation. At time $t = 5$ s, the On-Off TCP flows are activated and remain open for a duration of 10 s. The ON/OFF periods of the non-persistent connections are derived from Pareto distributions with a shape equal to 1.3. The mean ON period and the mean OFF period are set to 160 ms, a value, on the average, equal to the flows RTT. This choice is deliberately made to prevent the ON/OFF TCP sources from entering the slow-start phase even after periods of idleness. This helps to illustrate the resiliency of the REFWA even in the case of significant amount of burstiness in the network.

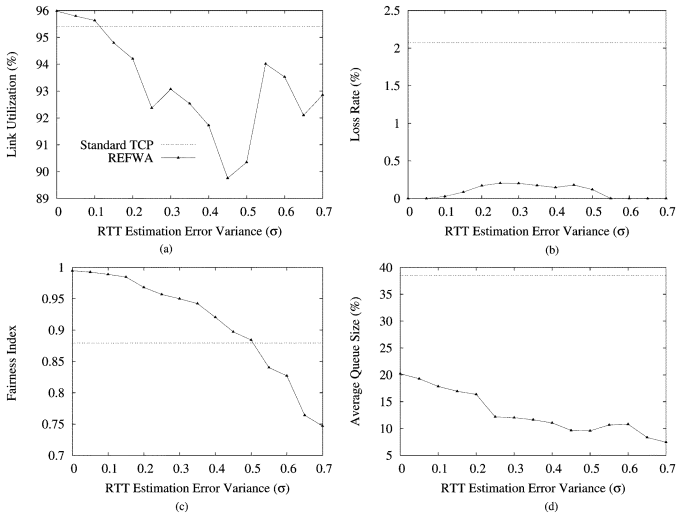


Fig. 7. Resiliency to errors in RTT estimation for different error variances ($\phi = 1.5$, $\psi = 0.5$, $\alpha = 1$). (a) Link utilization (%). (b) Loss rate (%). (c) Fairness index. (d) Average queue size (%).

Fig. 8 shows the bottleneck utilization, drops rate, fairness index values, and average queue size for different numbers of ON/OFF flows count. The results demonstrate the good performance of the REFWA scheme in environments with bursty traffic. The scheme maintains a higher utilization of the bottleneck link and significantly reduces the number of drops even for higher number of ON/OFF flows. This is because unlike standard TCP, the REFWA algorithm attempts to bound the aggregate window sizes of all active TCP flows to the bandwidth–delay product of the network and thus avoids overloading the bottleneck link with packets. The sequence number growths of the ten persistent TCP connections and some of the ON/OFF flows are plotted in Fig. 9. Since the number of ON/OFF TCP sources is large (case of 60 ON/OFF flows), the sequence numbers of only source 15 and every other fifth source thereafter are plotted. The figure confirms the ability of the REFWA scheme to achieve fairness even in the presence of

[4]The difference between the erroneous and good estimates of RTT.

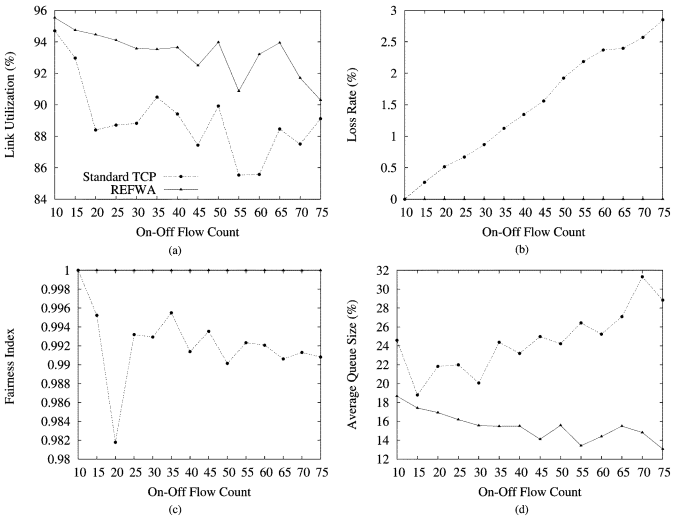


Fig. 8. Overall performance for different values of On-Off flow counts ($\phi = 1.5$, $\psi = 0.5$, $\alpha = 1$). (a) Link utilization (%). (b) Loss rate (%). (c) Fairness index. (d) Average queue size (%).

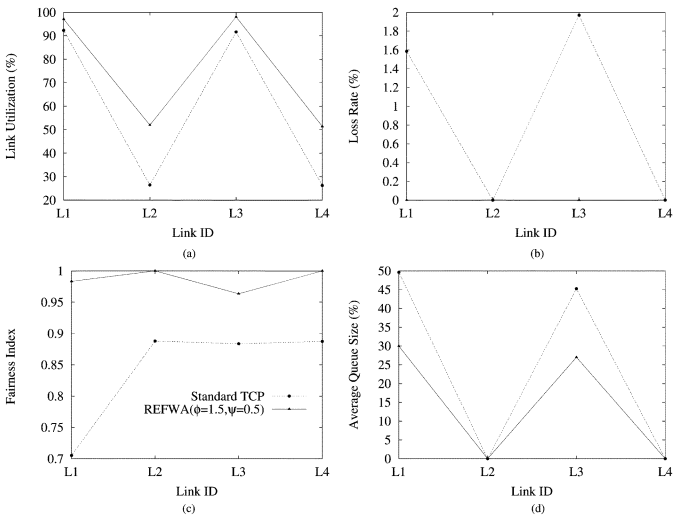


Fig. 10. Overall network performance of the four links ($\phi = 1.5$, $\psi = 0.5$, $\alpha = 1$). (a) Link utilization (%). (b) Loss rate (%). (c) Fairness index. (d) Average queue size (%).

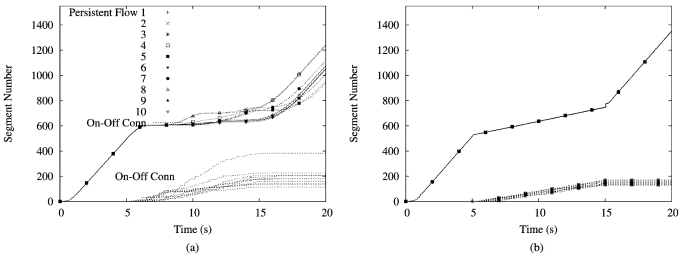


Fig. 9. Sequence number growth of both persistent and on-off connections (case of On-Off flow count = 60, $\phi = 1.5$, $\psi = 0.5$, $\alpha = 1$). (a) TCP. (b) REFWA.

traffic bursts. With the REFWA scheme, the progress of all persistent TCP connections remains fair during the running time of the simulation. Observe how the slope of the sequence number lines changes at time $t = 5$ s and $t = 15$ s as a result of entry and departure of ON/OFF flows, respectively. Note also that in case of only standard TCP, both ON/OFF and persistent flows exhibit great deviations, whereas REFWA helps all flows to fairly progress and to behave in an identical way. These results show that the REFWA scheme does not penalize the ON/OFF traffic for being idle and thus demonstrate the resiliency of the scheme to accommodate such dynamic traffic.

F. Performance in Presence of Multiple Bottlenecks

So far, a simple network environment has been considered. It has been assumed that only one bottleneck is traversed by many connections and that all flows are always making full use of their allocated windows. The remainder of this section envisions a general case where connections traverse multiple bottlenecks. Fig. 1(c) shows the example network configuration used in this study. The abstract configuration consists of three groups of flows. Group 1 consists of three flows and shares the first bottleneck (link *L1*) with the two flows of Group 2. Group 3 is formed of four flows and shares bandwidth of the second bottleneck (link *L2*) with flows of Group 1. ISL links are given a capacity of 1.544 Mb/s *(T1)*.

Fig. 10 graphs the link utilization, the drops rate, and the average queue size experienced by the four simulated inter-satellite links. The figure shows also the performance of the proposed scheme in terms of fairness. As explained previously, the feedback value of a connection can only be downgraded along its path. If the feedback value set by a node exceeds the value computed by its downstream counterpart, the feedback value is marked down to the smaller value. In case of the considered network topology, TCP connections of Group 1 are assumed to send data with rates fed back by link *L3* that are smaller than the windows allocated by link *L1*. This obviously compels flows of Group 1 not to fully utilize their allocated bandwidths at link *L1* causing the spare bandwidth to increase and the buffer occupancy to go down. This eventually leads to an increase in the computed feedbacks at link *L1*. Flows of Group 2 will transmit data at rates equal to the computed feedback and this helps to fully utilize the link capacity while maintaining small buffer sizes. Results of Fig. 10 demonstrate the high utilization of both links *L1* and *L2*, and confirm this observation. It is noticed that without REFWA, link *L1* is also highly utilized but this is at the price of system fairness. This is because flows of Group 2 conquer most of the link bandwidth given their short RTTs. Observe also the high number of losses and large queue sizes experienced when TCP connections are controlled by only standard TCP.

Figs. 11 and 12 present the queue occupancy and the average throughput of the bottleneck links *L1* and *L3*, respectively. The two figures actually confirm the results of Fig. 10. In fact, in case of TCP, both bottlenecks exhibit frequent oscillations in the queue occupancy and irregular fluctuations of the link average utilization. These transient behaviors cause a large number of packet drops and frequent idling of the bottleneck links, thereby resulting in lower link utilizations and higher drop rates. With the REFWA scheme, the link utilization of both bottlenecks

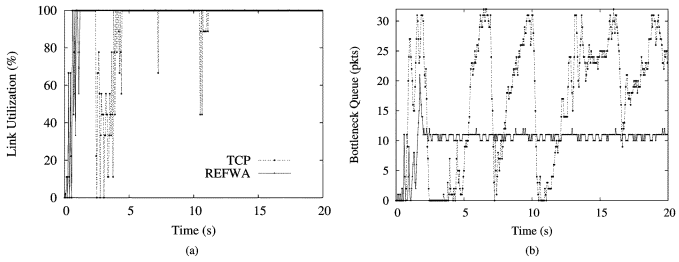


Fig. 11. Bandwidth utilization and queue size of link L1 averaged over 100ms time interval ($\phi = 1.5, \psi = 0.5, \alpha = 1$). (a) Bottleneck link utilization. (b) Bottleneck queue.

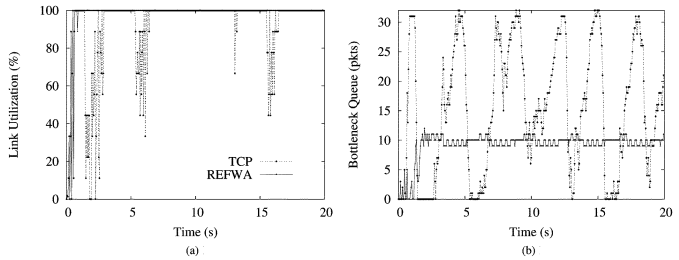


Fig. 12. Bandwidth utilization and queue size of link L3 averaged over 100 ms time interval ($\phi = 1.5, \psi = 0.5, \alpha = 1$). (a) Bottleneck link utilization. (b) Bottleneck queue.

is always near total capacity, and their queue sizes are small and exhibit limited oscillations. This explains the absence of drops and higher link utilization of both links in case of REFWA (Fig. 10).

G. Performance in Environments With Channel Errors

To compare the performance of the proposed scheme to that of Standard TCP based on the number of packet drops and overall link utilization, it has been presumed that all links were error-free in the simulations conducted so far. This assumption was made so as to avoid any possible confusion between throughput degradation due to packet drops and that due to satellite channel errors.

Admittedly, satellite environments are well known for their high bit error rates (BERs). These errors significantly impair the performance of TCP in satellite networks. To investigate the performance of REFWA in environments with BER, we consider the network configuration of Fig. 1(a). The used access link type is T2. The number of simulated flows is 10. Link errors are randomly derived from a uniform distribution based loss model, and are applied to up-links. The loss probability, expressed in packet unit and referred to as the packet error rate (PER), is varied within the range $[10^{-5}; 0.5]$. Fig. 13 shows the bottleneck link utilization and the fairness index in case of using REFWA and TCP for different PER rates, respectively.

The figures demonstrate that for lower PER rates, the REFWA scheme outperforms TCP and achieves better utilization of network resources and higher fairness. This performance is due to the fact that packet losses due to link errors are rare, and most of packet drops are due to network congestion. This is consistent with the simulation results presented above. On the other hand, for high PER errors, the two schemes experienced an abrupt

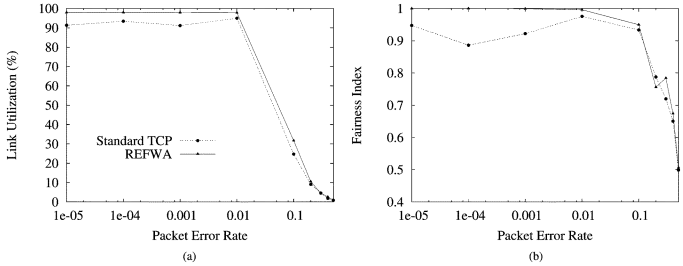


Fig. 13. Performance evaluation REFWA and standard TCP for different PER ($\phi = 1.5, \psi = 0.5, \alpha = 1$). (a) Average link utilization. (b) Fairness.

decrease in their link utilization and fairness index values. Effectively, we observe that in significantly higher PER environments (e.g., $\mathrm{PER} \geq 0.1$), the link utilization experienced in case of both REFWA and TCP is almost null. The low link utilization of REFWA and TCP is due to the fact that senders misinterpret packet losses as network congestion and halve their window sizes sometimes multiple times due to multiple losses within one window of data. In deed, in case of heavy errors, the slow start threshold (ssthresh) tends to take values (say one to four packets) significantly smaller than the optimal values. Reducing the congestion window (cwnd) to the ssthresh value, as in normal TCP, drastically throttles the TCP throughput. The throughput degradation becomes further significant as the idle waiting time becomes longer due to the retransmission time-out (RTO) backoff algorithm. To conclude, while REFWA gets its throughput degraded in high PER environments, it maintains a good performance in environments with PER less than 0.01, an error rate value unlikely to occur given the recent and ongoing advances in satellite channels.

Finally, in addition to the presented experiments, comparison between the performance of REFWA and the eXplicit and Fair Window Adjustment (XFWA) scheme proposed in [26] was also made and the better performance of REFWA in the case of both multiple-bottlenecks and single-bottleneck environments was verified. Due to the paper length limitation, we are not presenting these simulations results.

VI. Implementation Issues

It should be emphasized that there are several implementation issues that must be resolved when applying the REFWA scheme to practice. For instance, a REFWA satellite should be capable of reading and modifying the TCP header in packets. This violates the IP-Security (IPSec) semantics according to which packets must be processed by only the end-systems; no third party is allowed to modify the payload. Despite such violations, a number of schemes requiring flow identification and using window adjustment have been proposed in literature [22], [23].

The REFWA scheme could be costly in terms of resources. For example, since a REFWA satellite maintains state information for individual TCP flows, this will obviously incur more overheads at the system in terms of memory, few bytes per flow. However, the obtained performance gains (higher fairness, higher link utilizations, smaller queue sizes, and lower packet loss rates) are considerable and can be used to justify this additional cost. As for the feedback computation, it is fairly simple and requires only a few additions and a few multiplications every $\mathrm{RTT}_{\mathrm{avg}}$ interval of time, as explained in Appendix II.

So far, we have considered end-terminals connected directly to the satellite network. We can, however, extend our studies to more general scenarios where users are placed in a fixed Internet and are connected to the satellite network via terrestrial gateways. In such case, we adopt the TCP-split concept [15] used in many pioneering research work (e.g., PETRA [38]). The overall communication path between two end-terminals can be split into three connections, namely one between the sender and its correspondent gateway, another between the receiver and its correspondent gateway, and the third between the two gateways. Application of the proposed scheme is then relevant to only the latter.

Another concern about the REFWA is that packets and their acknowledgments must follow the same path if estimate of flows RTT and TCP's advertised window adjustment are to be effective. Indeed, in satellite networks, when the connection path is longer than one hop, more than one possible path between the end-systems can be simultaneously used. These multiple paths cause both data packets and acknowledgments to be received out of order and eventually degrade the overall throughput of the connection. One suggested approach to tackle this problem of in-order delivery in multi-path environments is the scheme proposed in [39]. The scheme measures the total propagation delays and queueing delays of these multiple paths, and uses this information to select an optimal and single path for each connection. The implementation of such a scheme can help REFWA satellites to compel forward traffic and the corresponding backward traffic to travel along the same route.

Another scenario where data packets and acknowledgments may travel along different paths and might eventually impose limitations on the performance of REFWA occurs when either the destination or the source undergoes handover from one satellite to another. Upon handover occurrence, routing tables are updated, and packets that are still in transit to the satellite that was being used by the end-system (before handover occurrence), will be routed onward to the current satellite the end-system is now using. This will cause the "in-transit" packets to travel one extra hop and eventually results in an abrupt increase or decrease in the flow delay. As a consequence, erroneous estimates of flows RTT and the total number of flows may be obtained. Feedback values of all TCP flows might accordingly be affected. Nonetheless, since parameters are periodically updated, handover occurrence is most unlikely to coincide with the time of the update operation. Even if such a coincidence happens, feedbacks are periodically computed every $\mathrm{RTT}_{\mathrm{avg}}$ interval of time and good estimates of parameters can be obtained in the next update interval of time. The effect of these "in-transit" packets will be thus minimal.

VII. Conclusion

In this paper, we proposed a recursive, explicit, and fair congestion control method specifically designed for multi-hops satellite constellations. The proposed method improves TCP performance in LEO satellite networks and represents a major contribution in the area of TCP performance over satellite networks.

The method takes advantage of some specific attributes of multi-hops satellite constellations to make an approximate estimate of flows RTT and the bandwidth–delay product of the network. To control network utilization, the scheme matches the sum of window sizes of all active TCP connections sharing a bottleneck link to the effective network bandwidth–delay product. On the other hand, Min-max fairness is achieved by assigning for each connection a weight proportional to its round-trip time. The computed feedbacks are signaled to TCP senders by modifying the receiver's advertised window (RWND) field carried by TCP ACKs. This operation can be accomplished without changing the protocol and, as a result, requires no modification to the TCP implementations in the end systems. To make the proposed method independent of the average RTT of TCP flows so the system would be applicable to any network topology with no further tuning, the stability of the REFWA scheme was studied and its parameters were set to fixed values.

We demonstrated through extensive simulation results that REFWA has the potential to substantially improve the system fairness, reduce the number of losses, and make better utilization of the link. Experiments with dynamic changes in traffic demands showed that REFWA managed to control the buffer occupancy well and to achieve stability when a change in traffic load occurred. A large part of this success is due to the ability of REFWA to rapidly divide the available bandwidth among all active flows while backing off the transmission rate of old flows upon arrival of new connections. Simulations of flows with different RTTs demonstrated the robustness of REFWA to high variance in the flows RTT distribution and showed that REFWA is significantly fair and has no bias against long RTT flows. Resiliency of the scheme to reasonable RTT estimation errors was demonstrated by simulation results and its ability to accommodate bursty traffic was verified also by considering a scenario where a mix of greedy and non-persistent flows were competing for the bandwidth of the bottleneck link. A multiple-bottlenecks network was considered and the good performance of REFWA in such environment was verified by simulation results. The performance of the REFWA scheme was also evaluated in environments with channel errors. The simulation results indicated that the proposed scheme maintained its good features even in the presence of reasonable packet error rates.

Appendix I
Stability Analysis of the Feedback Control Scheme

As the feedback computation is performed in a recursive manner and the feedback equation represents a close-loop system-control equation, stability in such case is required. Stability refers to the fact that under any conditions (e.g any number of flows and any distribution of their RTTs), the system should always converge to its point of equilibrium; transmission of the highest number of packets without causing network congestion. This appendix discusses the stability of the proposed scheme through a mathematical analysis. From (2), the aggregate TCP window size at time $(t = n \cdot \mathrm{RTT}_{\mathrm{avg}})$ is expressed as

$$\Upsilon(n) = \Upsilon(n-1) + \phi\Big(Bw \cdot \mathrm{RTT}_{\mathrm{avg}} - \Upsilon(n-1)\Big) + \psi\Big(B - Q(n-1)\Big). \quad (3)$$

Noticing that $Q(n-1)$ is the queue size resulted from unsent packets of windows $W(n-2)$, $Q(n-1)$ can be expressed as follows:

$$Q(n-1) = \Upsilon(n-2) - Bw \cdot \mathrm{RTT}_{\mathrm{avg}}.$$

We then obtain the following system-control equation:

$$\Upsilon(n) + \Omega_1 \cdot \Upsilon(n-1) + \Omega_2 \cdot \Upsilon(n-2) = \Omega_3 \tag{4}$$

where the triplet

$$\begin{aligned}\Omega_1 &= \phi - 1\\ \Omega_2 &= \psi\\ \Omega_3 &= \psi \cdot B + Bw \cdot \mathrm{RTT}_{\mathrm{avg}} \cdot (\phi + \psi).\end{aligned}$$

The equation can be simplified to

$$X(n) + \Omega_1 X(n-1) + \Omega_2 X(n-2) = 1$$

where

$$X(n) = \Upsilon(n) + \frac{1-\Omega_3}{1+\Omega_1+\Omega_2}.$$

Using the z-transform method, the discrete open-loop transfer function of the system is as follows:

$$\begin{aligned}Z\{X(k)\} &= X_z(z)\\ X_z(z) &= \frac{z^3}{z-1} \cdot \frac{1}{z^2 + \Omega_1 z + \Omega_2}.\end{aligned}$$

Using Tustin's approximation

$$X_z(z) = X_s(s)_{s=\frac{2}{\mathrm{RTT}_{\mathrm{avg}}} \cdot \frac{1-z^{-1}}{1+z^{-1}}} \tag{5}$$

we obtain

$$X_s(s) = M_1 \cdot \frac{(s+M_2)^3}{s(M_3 s^2 + M_4 s + M_5)} \tag{6}$$

where the quintuplet

$$\begin{aligned}M_1 &= \frac{\mathrm{RTT}_{\mathrm{avg}}^2}{2}\\ M_2 &= \frac{2}{\mathrm{RTT}_{\mathrm{avg}}}\\ M_3 &= (\Omega_2 - \Omega_1 + 1)\mathrm{RTT}_{\mathrm{avg}}^2\\ M_4 &= 4(1-\Omega_2)\mathrm{RTT}_{\mathrm{avg}}\\ M_5 &= 4(\Omega_1 + \Omega_2 + 1).\end{aligned}$$

The system is stable if all the roots of the transfer function denominator polynomial have negative real parts. This condition is satisfied when

$$\begin{aligned}0 \le \Omega_1 &\le \Omega_2 + 1\\ \Omega_2 &\le 1.\end{aligned}$$

The magnitude and angle of the open-loop transfer function are

$$\begin{aligned}|Y_s| &= M_1 \frac{\sqrt{M_2^2 + w^2}^3}{w\sqrt{M_4^2 w^2 + (M_5 - M_3 w^2)^2}}\\ \angle Y_s &= \frac{-\pi}{2} + 3\mathrm{Arctan}\Big(\frac{w}{M_2}\Big) - \mathrm{Arctan}\Big(\frac{M_4 w}{M_5 - M_3 w^2}\Big).\end{aligned}$$

So as that the phase margin and magnitude are independent of the average RTT, we firstly put $(w = \kappa/\mathrm{RTT})$ where κ is a constant. We then have the equation shown at the bottom of the page. To simplify the system, we consider the case of $\kappa = 2$ which represents a case of a break point frequency, and compute Ω_1 and Ω_2 such that the crossover frequency is the same as the break point frequency, in other words, $|Y_s(\kappa = 2)| = 1$. After calculation, we obtain the following condition:

$$(1-\Omega_2)^2 + \Omega_1^2 = \frac{1}{2}. \tag{7}$$

A specific case is when $\phi = \frac{3}{2}$, $\psi = \frac{1}{2}$.

Appendix II
Feedback Computation Load

This appendix aims to demonstrate that REFWA is fairly simple to implement and does not require large computation work at the routers on-board the satellites.

First, let h_j denote the number of hops traversed by the flows of the jth group. From (1), the estimated RTT value of flows of the jth group is

$$\mathrm{RTT}_j = 2\hbar_j \cdot \mathrm{ISL}_{\mathrm{delay}}$$

where $(\hbar_j = h_j + 1)$. From (2), the feedback of flows of the mth group can be expressed as

$$F_j(n) = \hbar_j^{\alpha} \cdot \frac{1}{\sum_{k=1}^{M} n_k \hbar_k^{\alpha}} \cdot \Upsilon(n).$$

$$\begin{aligned}|Y_s| &= \frac{\sqrt{4+\kappa^2}^3}{2k\sqrt{16(1-\Omega_2)^2\kappa^2 + (4(\Omega_1+\Omega_2+1) - \kappa^2(\Omega_2-\Omega_1+1))^2}}\\ \angle Y_s &= \frac{-\pi}{2} + 3\mathrm{Arctan}\,(\kappa/2) - \mathrm{Arctan}\left(\frac{4\kappa(1-\Omega_2)}{4(\Omega_1+\Omega_2+1) - k^2(\Omega_2-\Omega_1+1)}\right)\end{aligned}$$

Let N denote the total number of TCP flows. The $\mathrm{RTT}_{\mathrm{avg}}$ and N are computed as follows:

$$\mathrm{RTT}_{\mathrm{avg}} = 2\frac{\sum_{k=1}^{M} n_k \cdot \hbar_k}{N} \cdot \mathrm{ISL}_{\mathrm{delay}}$$
$$N = \sum_{k=1}^{M} n_k.$$

Taking account of the $\mathrm{RTT}_{\mathrm{avg}}$ and N computations, and considering the case of $\alpha = 1$ (for the sake of simplicity), the computation of $\Upsilon(n)$ gives rise to only one division, two subtractions, $(2M + 2)$ additions, and $(M + 3)$ multiplications. The term $\left(\frac{1}{\sum_{k=1}^{M} n_k \hbar_k^{\alpha}}\right)$, on the other hand, requires only one division, M additions, and M multiplications. Since the values of $\Upsilon(n)$ and $\left(\frac{1}{\sum_{k=1}^{M} n_k \hbar_k^{\alpha}}\right)$ are the same for all flows, the periodic feedback computation of the whole system necessitates only two divisions, two subtractions, $(3M + 2)$ additions, and $(2M + 3)$ multiplications.

On the other hand, the value of M depends on the architecture of the satellite constellation and is always inferior than the maximum number of hops that can be traversed by any connection between any two terrestrial terminals. This threshold is denoted as χ. For instance, Wood [40] has shown through extensive simulation results that the longest one-way propagation delay experienced in the Teledesic constellation was less than 140 ms. Given the fact that the ISL delay is 20 ms, the parameter χ can be assumed to be six in case of Teledesic constellation.

To conclude, the implementation of REFWA scheme is fairly simple and routers on-board REFWA satellites are not required to perform large work to compute feedbacks: a maximum of two divisions, two subtractions, $(3\chi + 2)$ additions, and $(2\chi + 3)$ multiplications. Furthermore, this computation work is not performed per packets or flows, but only once every $\mathrm{RTT}_{\mathrm{avg}}$ interval of time. In case of Teledesic, the maximum amount of computation load required at each REFWA router is merely two divisions, two subtractions, 20 additions, and 15 multiplications. Note that this amount of computation load is practical even for high-speed routers.

ACKNOWLEDGMENT

The authors acknowledge the valuable comments and suggestions of the three anonymous reviewers who have greatly helped us improve the quality of this manuscript.

REFERENCES

[1] I. F. Akyildiz, E. Ekici, and M. D. Bender, "MLSR: A novel routing algorithm for multilayered satellite IP networks," *IEEE/ACM Trans. Netw.*, vol. 10, no. 3, pp. 411–424, Jun. 2002.

[2] K. Thompson, G. J. Miller, and R. Wilder, "Wide-area Internet traffic patterns and characteristics," *IEEE Network Mag.*, vol. 11, no. 6, pp. 10–23, Nov./Dec. 1997.

[3] R. Morris, "TCP behavior with many flows," in *Proc. IEEE Int. Conf. Network Protocol (ICNP'97)*, Atlanta, GA, Oct. 1997, pp. 205–211.

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

[5] Y. R. Yang, M. S. Kim, and S. S. Lam, "Transient behaviors of TCP-friendly congestion control protocols," in *Proc. IEEE INFOCOM*, Anchorage, AK, Mar. 2001, pp. 1716–1725.

[6] G. Montenegro, S. Dawkins, M. Kojo, V. Magretand, and N. Vaidya, "Long thin networks," RFC 2757, 2000.

[7] L. Wood, G. Pavlou, and B. Evans, "Effects on TCP of routing strategies in satellite constellations," *IEEE Commun. Mag.*, vol. 39, no. 3, pp. 172–181, Mar. 2001.

[8] T. Taleb, N. Kato, and Y. Nemoto, "A recursive, explicit and fair method to efficiently and fairly adjust TCP windows in satellite networks," in *Proc. IEEE Int. Conf. Communications (ICC)*, Paris, France, Jun. 2004, pp. 4268–4274.

[9] C. Partridge and T. J. Shepard, "TCP/IP performance over satellite links," *IEEE Network Mag.*, vol. 11, no. 5, pp. 44–49, Sep./Oct. 1997.

[10] M. Allman, S. Floyd, and C. Partridge, "Increasing TCP's initial window," RFC 2414, 1998.

[11] I. F. Akyildiz, G. Morabito, and S. Palazzo, "TCP-peach: A new congestion control scheme for satellite IP networks," *IEEE/ACM Trans. Netw.*, vol. 9, no. 3, pp. 307–321, Jun. 2001.

[12] V. N. Padmanabhan and R. Katz, "TCP fast start: A technique for speeding up web transfers," in *Proc. IEEE GLOBECOM*, Sydney, Australia, 1998, pp. 41–46.

[13] T. Henderson and R. Katz, "Transport protocols for Internet-compatible satellite networks," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 2, pp. 326–344, Feb. 1999.

[14] H. Balakrishnan, V. N. Padmanabhan, and R. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," *IEEE/ACM Trans. Netw.*, vol. 5, no. 6, pp. 756–769, Dec. 1997.

[15] A. Bakre and B. R. Badrinath, "I-TCP: Indirect TCP for mobile hosts," in *Proc. 15th Int. Conf. Distributed Computing Systems (ICDCS'95)*, Vancouver, Canada, May 1995, pp. 136–143.

[16] S. Athuraliya, V. H. Li, S. H. Low, and Q. Yin, "REM: Active queue management," *IEEE Network Mag.*, vol. 15, no. 3, pp. 48–53, May 2001.

[17] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Netw.*, vol. 1, no. 4, pp. 397–413, Aug. 1993.

[18] K. K. Ramakrishnan and S. Floyd, "A proposal to add Explicit Congestion Notification (ECN) to IP," RFC 2481, 1999.

[19] S. H. Low, F. Paganini, J. Wang, S. Adlakha, and J. C. Doyle, "Dynamics of TCP/AQM and a scalable control," in *Proc. IEEE INFOCOM*, New York, Jun. 2002, pp. 239–248.

[20] L. S. Brakmo and L. L. Peterson, "TCP Vegas: End to end congestion avoidance on a global Internet," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 8, pp. 1465–1480, Oct. 1995.

[21] D. Katabi, M. Handley, and C. Rohrs, "Congestion control for high bandwidth-delay product networks," in *Proc. ACM SIGCOMM*, Pittsburgh, PA, Aug. 2002, pp. 89–102.

[22] L. Kalampoukas, A. Varma, and K. K. Ramakrishnan, "Explicit window adaptation: A method to enhance TCP performance," *IEEE/ACM Trans. Netw.*, vol. 10, no. 3, pp. 338–350, Jun. 2002.

[23] J. Aweya, M. Ouellette, D. Y. Montuno, and Z. Yao, "WINTRAC: A TCP window adjustment scheme for bandwidth management," *Perform. Eval.*, vol. 46, no. 1, pp. 1–44, Sep. 2001.

[24] A. K. Choudhury and E. L. Hahne, "Dynamic queue length thresholds in a shared memory ATM switch," in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar. 1996, pp. 679–687.

[25] T. Taleb, N. Kato, and Y. Nemoto, "On improving the efficiency and fairness of TCP over broadband satellite networks," in *Proc. IEEE VTC 2003 Fall*, Orlando, FL, Oct. 2003.

[26] ——, "An explicit and fair window adjustment method to enhance TCP efficiency and fairness over multi-hops satellite networks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 2, pp. 371–387, Feb. 2004.

[27] C. Jin, H. Wang, and K. G. Shin, "Hop-count filtering: An effective defense against spoofed traffic," in *Proc. 10th ACM Int. Conf. Computer and Communications Security (CCS)*, Washington, DC, Oct. 2003, pp. 30–41.

[28] R. Morris, "Scalable TCP congestion control," in *Proc. IEEE INFOCOM*, Tel Aviv, Israel, Mar. 2000, pp. 1176–1183.

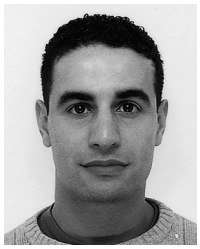
[29] S. B. Fredj, S. O. Boulahia, and J. W. Robers, "Measurement-based admission control for elastic traffic," in *Proc. Int. Teletraffic Congress (ITC-17)*, Salvador da Bahia, Brazil, Sep. 2001, pp. 161–172.

[30] L. Qiu, Y. Zhang, and S. Keshav, "On individual and aggregate TCP performance," in *Proc. 7th Annu. Int. Conf. Network Protocols (ICNP)*, Toronto, Canada, Oct./Nov. 1999.

[31] UCB/LBNL/VINT Network Simulator—ns (Version 2), VINT Project [Online]. Available: http://www.isi.edu/nsnam/ns/

[32] S. Floyd and T. Henderson, "The NewReno modifications to TCP's Fast Recovery Algorithm," RFC 2582, 1999.

[33] R. Goyal and R. Jain, "Buffer management and rate guarantees for TCP over satellite-ATM networks," *Int. J. Satellite Commun.*, vol. 19, no. 1, pp. 111–129, 2001.
[34] D. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Compute Networks and ISDN Systems*, vol. 17, pp. 1–14, Jun. 1989.
[35] J. Kulik, R. Coulter, D. Rockwell, and C. Partridge, "Paced TCP for high delay-bandwidth networks," in *Proc. IEEE GLOBECOM*, Rio de Janeiro, Brazil, Dec. 1999.
[36] K. Park, G. Kim, and M. Crovella, "On the relationship between file sizes, transport protocols, and self-similar network traffic," in *Proc. IEEE Int. Conf. Network Protocols (ICNP '96)*, Columbus, OH, Oct. 1996, pp. 171–180.
[37] W. Willinger, M. Taqqu, R. Sherman, and D. Wilson, "Self-similarity through high variability: Statistical analysis of ethernet LAN traffic at the source level," in *Proc. ACM SIGCOMM*, Cambridge, MA, Aug. 1995, pp. 100–113.
[38] M. Marchese, M. Rossi, and G. Morabito, "PETRA: Performance enhancing transport architecture for satellite communications," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 2, pp. 320–322, Feb. 2004.
[39] M. L. Liron, "Traffic routing for satellite communication system," U.S. Patent 5,740,164, Apr. 14, 1998.
[40] L. Wood, "Internetworking with satellite constellations," Ph.D. thesis, Univ. Surrey, Surrey, U.K., 2001 [Online]. Available: http://www.ee.surrey.ac.uk/Personal/L.Wood/publications/PhD-thesis/



Tarik Taleb (M'05) received the B.E. degree in information engineering with distinction, and the M.E. and Ph.D. degrees in computer sciences from the Graduate School of Information Sciences, Tohoku University, Japan, in 2001, 2003, and 2005, respectively.

He is currently working as an Assistant Professor with the Graduate School of Information Sciences, Tohoku University. From October 2005 until March 2006, he was working as a Research Fellow with the Intelligent Cosmos Research Institute, Sendai, Japan. His research interests lie in the field of wireless networking, satellite and space communications, congestion control protocols, mobility management, and network security.

Dr. Taleb is on the editorial board of the *IEEE Wireless Communications Magazine*. He serves also as Secretary of the Satellite and Space Communications Technical Committee of the IEEE Communication Society (ComSoc). He has been on the technical program committee of several IEEE conferences, including Globecom, ICC, and WCNC, and has chaired some of their sessions. He has acted as reviewer for many IEEE conferences, *IEICE Transactions on Communications*, and IEEE/ACM TRANSACTIONS ON NETWORKING. He is a recipient of the Niwa Yasujirou Memorial award (February 2005) and the Young Researcher's Encouragement award from the Japan chapter of the IEEE Vehicular Technology Society (VTS) (Oct. 2003).

Nei Kato (SM'05) received the M.S. and Ph.D. degrees from the Graduate School of Information Sciences, Tohoku University, Sendai, Japan, in 1988 and 1991, respectively.

He has been working for Tohoku University since then and is currently a full Professor at the Graduate School of Information Sciences. He has been engaged in research on computer networking, wireless mobile communications, image processing, and neural networks.

He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan. He has served on the technical program and organizing committees of many international conferences.

Yoshiaki Nemoto (SM'05) received the B.E., M.E., and Ph.D. degrees from Tohoku University, Sendai, Japan, in 1968, 1970, and 1973, respectively.

He is a full Professor with the Graduate School of Information Sciences, and served as Director of the Information Synergy Center, Tohoku University. He has been engaged in research work on microwave networks, communication systems, computer network systems, image processing, and handwritten character recognition.

Prof. Nemoto was a recipient of the 2005 Distinguished Contributions to Satellite Communications award from IEEE Communications Society, and a co-recipient of the 1982 Microwave Prize from the IEEE MTT Society. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan and a Fellow of the Information Processing Society of Japan.