

# TCP Live Experiments on a Real GEO Satellite Testbed

C. Caini, R. Firrincieli, D.  
Lacamera  
DEIS  
University of Bologna, Italy  
{rfirrincieli,ccaini}@deis.unibo.it

T. De Cola, M. Marchese  
CNIT  
University of Genoa, Italy  
{mario.marchese,  
tommaso.decola}@cni.it

C. Marcondes,  
M. Y. Sanadidi, M. Gerla  
UCLA  
UCLA, Los Angeles, US  
{cesar,medy,gerla}@cs.ucla.edu

**Abstract** The paper describes a measurement campaign recently carried out by University of Bologna (UoB), National Inter-University Consortium for Telecommunications- (CNIT) and University of California Los Angeles (UCLA), on the CNIT GEO Skyplex platform. The aim of the experiments is the performance assessment of a wide range of TCP enhancements on network environments that include a real GEO satellite link. Measurements were carried out by means of the integrated UoB-CNIT testbed, fully controlled by remote through a web interface. Analysis of live experiments data confirm the challenges posed by GEO sat channels, and show a negative influence of the Skyplex BoD technique on TCP start-up performance. Among the TCP enhancements compared in the paper, best results are provided by TCP Hybla and TCP Westwood, with an advantage of the former in heterogeneous environments characterized by a high level of RTT unfairness.

**Index Terms** — Transport Protocols, Satellite Communications, Testbed, Measurements, TCP Westwood, TCP Hybla.

## I. INTRODUCTION

The paper describes a measurement campaign recently carried out by University of Bologna (UoB), National Inter-University Consortium for Telecommunications- (CNIT) and University of California Los Angeles (UCLA) on the CNIT GEO Skyplex platform [1]. As widely recognized [2], satellite channels are really challenging for transport protocol performance, because of long Round Trip Times (RTT) (about 600 ms for GEO) and the possible presence of random errors (depending on satellite platform characteristics and propagation conditions). To overcome these problems, intermediate agents on the path, usually called Performance Enhancing Proxies (PEPs), can be envisaged [3], [4], [5]. Alternatively, to avoid PEP drawbacks (in particular the violation of the end-to-end TCP semantics in “TCP

splitting” PEPs), better performing TCP variants can be considered as well. This paper focuses on approach, aiming to analyze performance of a wide range of TCP enhancements on environments that include a real GEO satellite link.

The scenarios investigated can be distinguished into two categories: pure satellite networks and heterogeneous networks. In the first case only satellite connections (i.e. TCP connections that include the satellite leg) are considered. In the latter, both satellite and pure wired connections (i.e. TCP connections whose legs are all wired) are present. This second environment, which is quite typical of Internet communications, is the most demanding, because of the severe RTT-unfairness against long RTT satellite connections. Measurements were carried out by means of the integrated UoB-CNIT testbed [6], [7], whose logical layout is sketched in and which can be fully controlled by remote through its web interface. The TCP variants considered are: SACK [8], BIC [9], Vegas [10], Hybla [11], Westwood plus [12] and Westwood [13]. The Multi-TCP [14] Linux kernel patch was instrumental for both the selection of the TCP variants and the collection of experiment results.

## II. TESTBED DESCRIPTION

The testbed consists of the following three logical and geographical sections (Figure 2):

- the testbed controller, located at UoB,;
- the UoB-CNIT integrated testbed, located at Genoa, composed by 5 Linux PCs and 1 FreeBSD;
- the Skyplex GEO satellite system and 1 Linux PC satellite receiver, located in Naples.

### A. The testbed controller

The controller, connected to the testbed via a Virtual Private Network, allows the user a ubiquitous remote control of the testbed via a standard web browser. In addition to this important feature it provides the following additional advantages:

This work was supported in part by the IST-027323 SatNEx II Network of Excellence.

- to conceal the software and hardware complexity of testbed;
- to improve security, as the user must be granted only a standard access to the testbed web page;
- to facilitating test scheduling,;
- to greatly speed up both test execution and result collection.

### B. *The integrated testbed*

The UoB-CNIT integrated testbed consists of several PCs running the Linux operating system, and a PC running FreeBSD (devoted to TCP Westwood and directly maintained by UCLA). Multi-TCP on Linux senders allows the user to easily select at runtime the TCP variants available on Linux kernel 2.6.12 and also implements the full version of TCP Hybla (including packet spacing and Hoe's initial ssthresh estimation).

Wireless connections are composed of wired legs and a final satellite link, while TCP background traffic is present only in the entirely wired paths. All the connections share the R1-R2 link (Figure 1 and 2), whose bandwidth can be limited in order to study congestion effects. The router R1 at its input can follow either a DT (Drop Tail) policy or a RED (Random Early Detection) policy. In both cases, the router parameters are fully configurable from the web interface.

The satellite connections can be realized either by emulation or, as of interest here, by the third testbed component, described in the following paragraph.

### C. *The Skyplex platform*

The DATA Skyplex platform [1] is based on the Eutelsat HotBird 6 satellite, which carries four Ka band transponders equipped with a Skyplex Unit. The units are composed of channels configurable in Low Rate (at 2.112 Mbps) or in High Rate (at 6.226 Mbps). The channels can be configured in SCPC or TDMA mode, capacity sharing only being available in the second case. The TDMA mode has a frame structure composed by  $N$  time slots, occupied by bursts, and as each burst contains  $M$  MPEG cells, we have  $N \times M$  cells per frame. In such a structure,  $N$  also represents the number of users per frame and defines the Skyplex TDMA Mode. In order to avoid the heavy constraint of having only up to  $N$  users sharing the channel, i.e. one burst per frame assigned to each terminal, the frame structure in the ViaSat system has been extended by grouping together  $L$  frames in a multi-frame. Assigning, as a minimum, one burst per multi-frame to each user, the number of users per channel becomes  $N \times L$ , with a minimum granularity in our system of 44 kbps ( $N=6, M=8, L=8, \text{Low Rate}$ ) instead of 352 kbps

( $N=6, M=8, L=1, \text{Low Rate}$ ) for a single frame structure. In fact, the burst configuration present on the satellite platform referenced in this paper corresponds to  $N=6, M=8, L=8$ . Finally, 3 multi-frames make up the super-frame defined for signalling purposes. The super frame length results in our configuration of 820ms, which is the period of retransmission and of validity of the schedule for the assignment of the multi-frame time slots to the terminals. Such a schedule is called the Burst Time Plan (BTP).

Bandwidth assignment can be either dynamic (Bandwidth on Demand, BoD), or static (Committed Information Rate, CIR) or mixed. In the first case, the bandwidth is requested periodically by each terminal on the basis of its own instantaneous need, and the time slots are assigned in a best-effort mode. The BoD assignment does not guarantee any priority between terminals or quality of service. In case of static allocation, a fix number of time slots is pre-configured and assigned to a particular terminal.

In particular, during our tests the system was working in dynamic bandwidth assignment, relying on RBDC (Rate Based Dynamic Capacity) [15], which consists in assigning bandwidth slots on the basis of the traffic data rate registered on the network interface (i.e. on the Skyplex modem buffer).

## III. PERFORMANCE METRICS AND MEASUREMENTS OBJECTIVES

The following metrics are considered in order to evaluate TCP performance over a real GEO satellite link:

- time sequence of transmitted packets; useful to study both connection efficiency and regularity;
- goodput, defined as the average amount of correctly received data (excluding retransmission) in the time unit; it is obtained by dividing the highest acknowledgment (ACK) received by the sender at the end of the connection by the connection length;
- short time goodput, defined as goodput before, except that correctly received data are averaged on a short-time sliding window; it allows studying the dynamics of performance (e.g. performance at start-up, or in presence of a non stationary concurrent traffic);

The aforementioned metrics are adopted in order to reach the following measurement objectives:

- goodput evaluation for different file transfer lengths;
- evaluation of start-up performance (time sequence of Tx packets and short time goodput);

- evaluation of capability to adapt to dynamic changes in the network traffic (short time goodput);
- evaluation of fairness between satellite and wired connections on a heterogeneous network (short time goodput and goodput).

Results presented in the evaluation sections are usually averaged over three identical tests, in order to cope with random effects introduced by the RED queue at bottleneck and by the Skyplex BoD technique.

#### IV. EVALUATION TOOLS

The adoption of Multi-TCP provides the user with the additional advantage of overcoming the limitations of standard Linux evaluation tools, as it collects from the kernel core a number of TCP variables, included the internal ones, such as the congestion window (cwnd) and the retransmission timeout (RTO). Alternative choices for TCP dynamics analysis range from coarse tcpdump to TCP web100 instrumentation [16].

For collecting data, Multi-TCP, inserts into the kernel TCP code three log routines, triggered by the following events:

- the encapsulation and passing of a TCP segment to the IP layer;
- the reception of a segment with the ACK flag active;
- the modification of the cwnd value.

The first routine reports the sequence number of the segment being sent out, its size, the total size of data sent out (excluding retransmissions), and some variables related to the TCP spacing algorithm. The second one gives the ACK number, the SND\_UNA and SND\_NXT values (as described in [17]), the estimated value of the “in flight” data, the receiver advertised window, the current state of the “Linux TCP/SACK/FACK/ECN engine state” [18] (OPEN, RECOVERY, etc.) and the three time variables, namely the mean RTT, the RTO and the RTT variance. The last one prints out the parameters related to the congestion control algorithm, i.e. the cwnd, the slow start threshold (ssthresh) and the maximum cwnd value (cwnd clamp).

All these data are collected in text files, which can be directly imported in a spreadsheet. Then, specifically implemented Excel macros make the data processing phase fast and easy, by automatically generating a set of default graphs [14].

#### V. PURE SATELLITE NETWORK

This scenario aims to evaluate the impact of satellite link characteristics on TCP performance, when the satellite connections are isolated from the other network components (i.e. in absence of wired cross traffic).

##### A. Link characterization

A basic characterization of a satellite link is usually given in terms of RTT and segment loss rate. Table 1 shows some RTT measures obtained by using the ping [19] tool.

In order to probe the available bandwidth of the satellite link and the segment loss rate due to radio impairments, persistent (600s) UDP flows have been adopted. UDP traffic has the important characteristics of being regularly spaced (inter-packet delay is constant at the sender side) and of not reacting to losses. Both features allow an estimation of the available bandwidth by simply increasing the Tx rate and observing at the receiver side how many packets are received. Measurements showed that UDP flows at 0.5 Mbps and 1 Mbps did not induce any loss, while 2 Mbps flow caused 48% of losses. The reasonable conclusion is that the physical layer error correcting code is able to assure a virtually loss-free satellite link, and that the observed losses should be ascribed only to congestion on the satellite gateway; consequently, for all the experiments we assume a null packet error rate and an available satellite bandwidth of 1 Mbps. Note that, in absence of concurrent traffic, the maximum goodput of satellite connections is limited to this value.

##### B. Single satellite connection

The aim of these tests is to assess start-up TCP performance and goodput for a single satellite connection.

###### 1) Start-up performance

Table 2 shows the time spent to reach and maintain an average goodput equal to the 90% of the link capacity for all the TCP variants at start-up. Figure 3 shows the time traces of Tx segments and ACKs with reference to the “best case” of three consecutive experiments. Lack of space force the authors to report results only for SACK, Westwood and Hybla (the first as a reference, the others because they provide better performance).

The outcome is that Hybla is the fastest in exploiting the link bandwidth, reaching the steady state before the other TCP variants (see Table 2). This is mainly due to the Hybla congestion control algorithm, which allows a fast opening of cwnd to counteract the effects of

large RTTs. Moreover, packet spacing and initial bandwidth estimation, allows Hybla (Figure 3, c) to avoid the early buffer overflow at the first router that affects other TCP variants (e.g. SACK, Figure 3, a), but Westwood (Figure 3, b). In spite of being the fastest, TCP Hybla performance at start-up falls, however, somewhat short of expected, and needs further verification. The possible explanations are discussed later. The good performance of Westwood is due to continuous update of a correct ssthresh value estimated during slow start, using the adaptive bandwidth share algorithm (ABSE) [20], which avoids the early buffer overflow, and provide a smoother convergence to the channel capacity. In spite of this advantage, Westwood does not have RTT-fairness features, so its performance appears slightly worse than Hybla, but better than SACK. The other TCP variants show intermediate results, but Vegas, which is greatly impaired by the satellite link characteristics. Vegas totally unsatisfactory performance is observed in all the experiments and it is very likely due to a harmful interaction between the BoD scheme of the Skyplex platform, which impacts the connection RTT variance [15], and the Vegas congestion estimation algorithm.

## 2) Goodput

Figure 4 shows the goodput of a single satellite connection in the case of different file transfer lengths, averaged over three consecutive experiments. In case of long file transfers, all of the TCP variants (but Vegas) succeed in exploiting the satellite link capacity (1 Mbps, estimated), also thanks to the large buffer at the satellite gateway. By contrast, in the case of short file transfers, all TCP variants are severely impaired by the large RTT and by the Skyplex BoD mechanism, which slows down the cwnd opening, in particular at start-up. Note that Hybla performance for short file transfer, although it is the best, is not as satisfactory as could be expected on the basis of previous investigation. This is likely due not only to the aforementioned BoD mechanism, but also to the advertised window of the Linux receiver, which does not grow fast enough to follow the Hybla start-up cwnd growth [21].

## VI. HETEROGENEOUS NETWORK

In this scenario satellite connections are no more isolated. Therefore, the focus is on the evaluation of the impact on the satellite connection of competing wired cross traffic.

### A. TCP reaction to UDP spike

In this case one satellite TCP connection runs for 180s. After the start-up phase is closed (about 60s), an UDP spike perturbation on the wired path (cross-traffic) starts and lasts for 10s, transmitting at 9.5 Mbps. This way, we effectively reduce the available bandwidth on the R1-R2 link to 0.5 Mbps,  $\frac{1}{2}$  of the satellite bandwidth. Note that with this choice during the UDP spike the bottleneck of satellite connection becomes the R1-R2 link. The capability of the satellite connection to react to the perturbation is the object of our investigation. To this end, for the same three TCP variants considered before, Figure 5 shows the short time goodput, calculated by adopting a sliding window of 10 s and by subsequently averaging over three consecutive experiments. Results show the superior agility of Hybla, due to its fast cwnd re-opening. The same conclusions are supported by Figure 6, which depicts the time requested to fully recover from the UDP perturbation (i.e. the time requested by the short time goodput to recover the value assumed at perturbation beginning), for some of the tested TCP variants.

### B. TCP behavior in case of cross traffic arriving

In this test, a satellite connection starts alone at time 0, followed by 5 wired connections (TCP SACK) when its start-up phase can be considered concluded (at about 60s). The capability of the satellite connection to maintain a satisfactory transmission rate in spite of the cross traffic arriving is investigated. Figure 7 a) shows that SACK is totally unable to maintain an adequate share of network resources to the satellite connection after the insertion of wired competitors. This is because of the large RTT disparity (600ms instead of 25 ms) which penalizes satellite connection [2]. Westwood (Figure 7 b) basically suffers of the same problem in the examined scenario, not being conceived to resolve the RTT disparity. It shows, however, more regular and better performance than SACK before competitor insertion. By contrast, Hybla (Figure 7 c) is able to maintain a short time goodput of about 500 kbps after the insertion of wired competitors. Even if this value is only a half of the available satellite link bandwidth, it is much higher than the performance achieved by the other variants considered, in the order of 100 kbps (Figure 7 a and b). As mentioned before, this behavior is mainly due to the capability of Hybla of fast re-opening the cwnd after a congestion event, in spite of large RTTs. However, note that the BoD technique prevents Hybla

from fully exploiting this feature, thus reducing its RTT-fairness capability. Finally, the other TCP variants (results are not reported here because of space constraints), by lacking explicit countermeasures against RTT unfairness, result unable to reach a satisfactory value of cwnd in presence of wired cross traffic.

### C. TCP behavior on entering a busy network

At the beginning of the test 5 wired connections (TCP SACK) are launched. Then, when the start-up phase can be considered concluded (at about 60 s), a satellite connection starts. The capability of the satellite connection to obtain a fair share entering an already congested network, i.e. starting from the most unfavorable conditions, is measured. Qualitative considerations done in the previous paragraph hold true also in this case. Results are summarized in Figure 8, where the goodput achieved by all TCP variants in this and in the previous case are reported.

## VII. CONCLUSIONS

Results obtained on live experiments confirm the severity of the challenges posed by GEO sat channels, as well as the negative impact of the Skyplex BoD technique on TCP start-up performance. Among the TCP enhancements compared in the paper, best results are provided by TCP Hybla and TCP Westwood, with a predominance of the former in heterogeneous environments. This result can be explained by considering that while TCP Hybla was specifically designed to counteract RTT-unfairness, TCP Westwood was mainly conceived to tackle random errors, which were not noticeably present during the experiments (based on fixed Tx and Rx platforms on clear sky and with well aimed antennas). Further research will be devoted to extend comparison to other satellite platforms and/or different environments, as well as to new TCP enhancements.

## REFERENCES

[1] [http://www.eutelsat.com/fr/satellites/pdf/dealers/annex\\_d\\_skyp lex.pdf](http://www.eutelsat.com/fr/satellites/pdf/dealers/annex_d_skyp lex.pdf)

[2] Y. Hu and V.O.H. Li, "Satellite-based internet: a tutorial", IEEE Commun. Mag., pp. 154-62, March. 2001.

[3] Border J, Kojo M, Griner J, Montenegro G, Shelby Z. "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations", IETF RFC 3135, June 2001

[4] C. Caini, R. Firrincieli, D. Lacamera, "PEPsal: a Performance Enhancing Proxy for TCP satellite connections", to appear on IEEE Aerospace and Electronic Systems Magazine

[5] C. E. Palazzi, C. Roseti, M. Luglio, M. Gerla, M.Y. Sanadidi and J. Stepanek, "Enhancing Transport Layer Capability in

HAPS--Satellite Integrated Architecture", Wirel. Pers. Commun., pp.339-356, Feb. 2005

[6] A. Bon, C. Caini, T. De Cola, R. Firrincieli, D. Lacamera, M. Marchese, "An Integrated Testbed for Wireless Advanced Transport Protocols and Architectures", in Proc. IEEE TridentCom06, Barcelona, Spain, March 2006, pp.522-525.

[7] Testbed web site: <https://deis57.deis.unibo.it>.

[8] M. Mathis and J. Mahdavi, "TCP Selective Acknowledgment Options", IETF RFC 2018, Oct. 1996.

[9] Lisong Xu, Khaled Harfoush, and Injong Rhee, "Binary Increase Congestion Control for Fast Long Distance Networks", in Proc IEEE INFOCOM '04, Hong Kong, March 2004, vol 4, 7-11 pp. 2514 – 2524.

[10] L. S. Brakmo and L. L. Peterson, "TCP Vegas: End to end congestion avoidance on a global internet", IEEE J. Select. Areas Commun., vol. 13, pp. 1465-1480, Oct. 1995.

[11] C. Caini, R. Firrincieli, "TCP Hybla: a TCP Enhancement for Heterogeneous Networks", Int. J. Satell. Commun. Network, vol. 22, pp.547-566, Sep.-Oct. 2004.

[12] L.A. Grieco and S. Mascolo, "Performance evaluation and comparison of Westwood+, New Reno, and Vegas TCP congestion control", ACM SIGCOMM Computer Commun. Review vol. 34, pp.25-38, April 2004.

[13] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, "TCP Westwood: end-to-end congestion control for wired/wireless networks", Wireless Networks, Kluwer Academic Publisher, vol.8, pp. 467-479, 2002.

[14] C. Caini, R. Firrincieli and D. Lacamera, "A Linux Based Multi TCP Implementation for Experimental Evaluation of TCP Enhancements", in Proc. SPECTS 2005, Philadelphia, July 2005, pp.875-883.

[15] A. Gotta, R. Secchi, and F. Potorti. An analysis of TCP startup over an experimental DVB-RCS platform, in Proc IWSSC06, pp. 176-180, Madrid (ES), September 2006..

[16] M. Mathis, J Heffner and R Reddy, "Web100: Extended TCP Instrumentation for Research, Education and Diagnosis", ACM Computer Communications Review, Vol 33, Num 3, July 2003.

[17] J. Postel, "Transmission control protocol", IETF RFC 793, Sept. 1981.

[18] P. Sarolahti and A. Kuznetsov, "Congestion Control in Linux TCP", in Proc. 2002 USENIX Annual Technical Conference, Freenix Track, Monterey, CA, June 2002, pp. 49–62.

[19] D.L. Mills, "Internet Delay Experiments", IETF RFC 889, December 1983.

[20] Ren Wang; Valla, M.; Sanadidi, M.Y.; Gerla, M., "Using adaptive rate estimation to provide enhanced and robust transport over heterogeneous networks," in Proc. IEEE ICNP 2002, pp. 206- 215, Nov.2002.

[21] <http://www.csm.ornl.gov/~dunigan/netperf/auto.html>

TABLE 1: RTT CHARACTERIZATION OF THE SATELLITE LINK

	min	avg	max	mean dev
RTT (ms)	<b>587.35</b>	<b>692.59</b>	<b>802.75</b>	<b>58.92</b>

TABLE 2: TIME TO REACH THE 90% OF THE SATELLITE LINK CAPACITY FOR A SINGLE SATELLITE CONNECTION.

TCP variant	Hybla	SACK	West wood	West wood+	BIC	Vegas
Time (s)	21	52	29	73	41	not reached

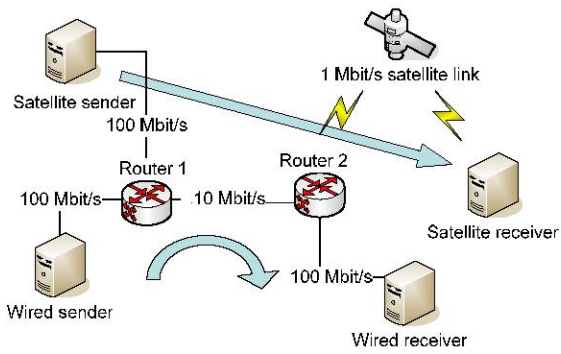


Figure 1: Testbed logical layout.

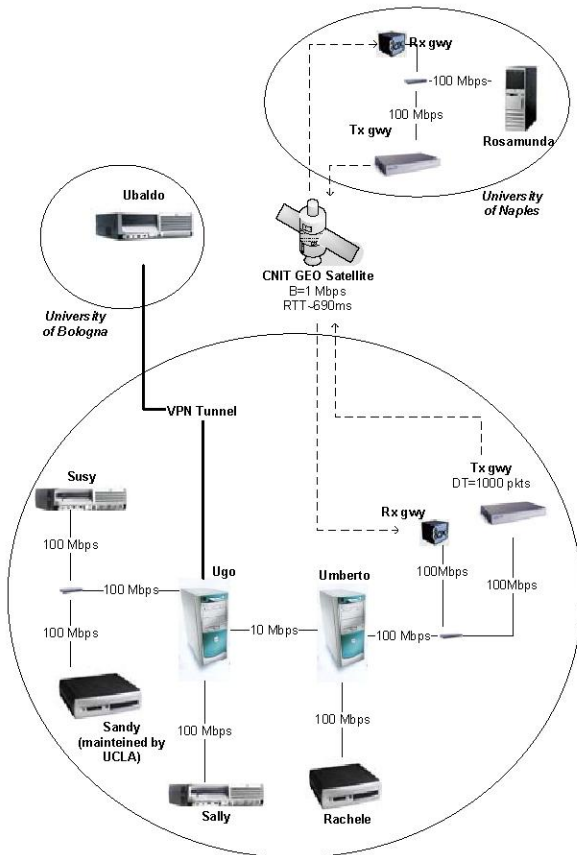


Figure 2: Testbed physical layout.

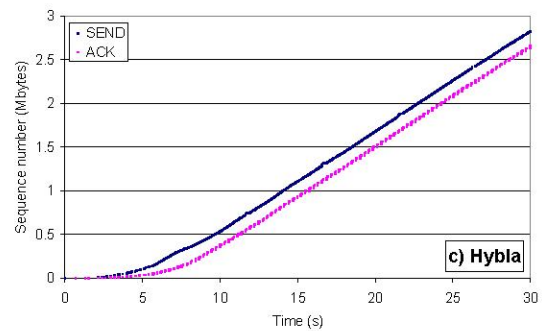
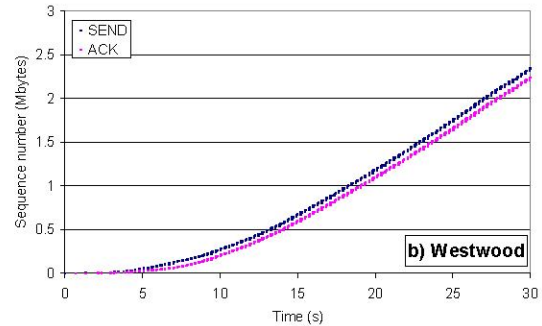
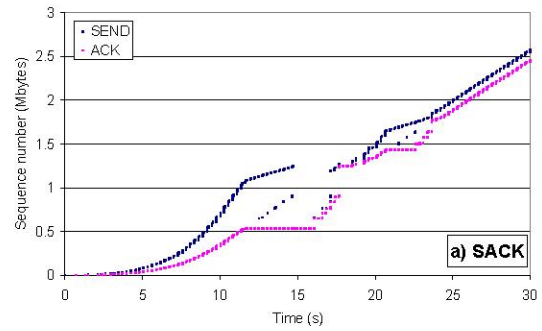


Figure 3: Transmitted segments and acknowledgments for Sack (a), Westwood (b) and Hybla (c) at start-up. Single test results (best case).

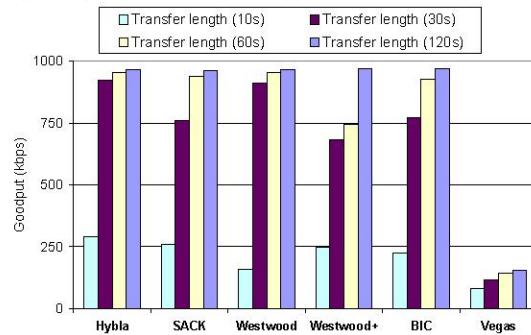


Figure 4: Goodput of a single satellite connection for each TCP variant and different file lengths. Average results over three tests.

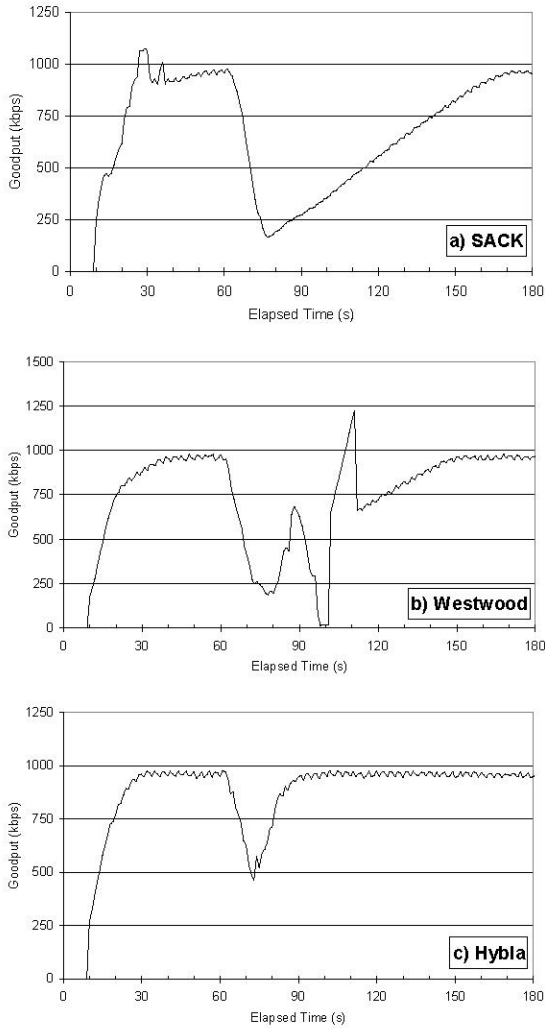


Figure 5: Short time goodput for Sack (a), Westwood (b) and Hybla (c) during the connection lifetime; a UDP spike perturbation starts at 60s lasting 10s. Average results over three tests.

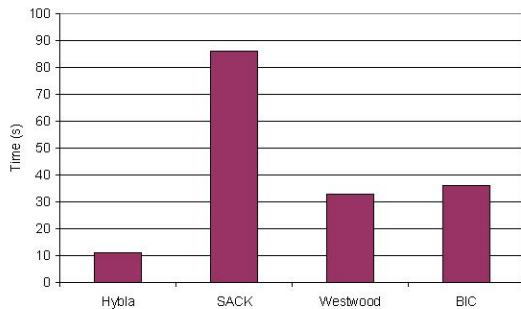


Figure 6: Average time requested by some TCP variants to recover from the UDP perturbation.

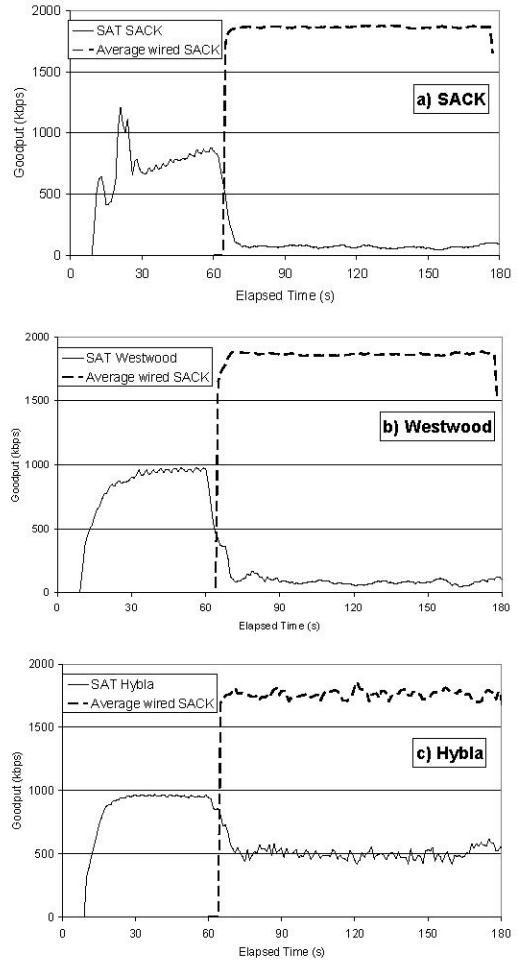


Figure 7: Sack (a), Westwood (b) and Hybla (c) behaviors for cross traffic arriving: short time goodput. Wired connections start 60s after sat connection. Average results over three tests.

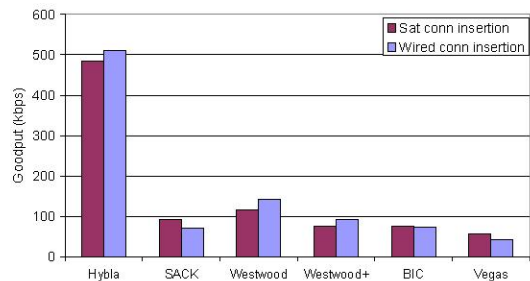


Figure 8: Goodput in presence of dynamic traffic for each TCP variant. Comparison between satellite traffic entering a bottleneck after and before wired cross traffic (Sat insertion and wired insertion, respectively). Average results over three tests.