Simulation Study of TCP Performance in TRANSAT Satellite Architecture

Davide Astuti
Department of Computer Science
University of Helsinki
davide.astuti@cs.helsinki.fi

Markku Kojo
Department of Computer Science
University of Helsinki
kojo@cs.helsinki.fi

ABSTRACT

The TRANSAT [1] satellite architecture is intended to provide efficient Internet access services over DVB-RCS satellite links. One of the key features in the TRANSAT architecture is the Satellite-Link Aware Communication Protocol (S-LACP), which operates at the link layer, and is employed over a satellite segment to decrease the residual packet error rate down to an acceptable level. In this simulation study, we measure the TCP performance over satellite links, including the effect of various TCP enhancements and the S-LACP protocol. We show that when S-LACP is used to reduce the residual packet rate as seen by the TCP layer, performance improves. Without S-LACP the modeled packet error rate tends to be rather high not allowing a TCP sender to open the congestion window large enough or later on to retain the congestion window fully open.

General Terms

Measurement, Performance, Design, Experimentation.

Keywords

TCP, satellite, performance.

1. INTRODUCTION

In recent years, the rapid global expansion of the Internet and the massive growth of wireless communication systems is leading towards a market where wireless data network and services would be predominant. Satellite systems are candidates to integrate the wireless data networks due to their wide coverage and broadcast abilities. However, the typical satellite links characteristics, such as high latency and high Bit Error Rate, affect the performance of the Internet transport protocols [2,3].

In this study, we measure the performance of several TCP enhancements selected for use with DVB-RCS (Digital Video Broadcasting-Return Channel System) satellite links in the TRANSAT [1] architecture, whose major goal is to provide efficient Internet access services over satellite links. DVB-RCS standard is an access technology, which allows Broadband access to the Internet and to value-added services via satellite. The leading guideline in the TRANSAT architecture is to follow the end-to-end design principle of the Internet architecture. A specific link-layer protocol, Satellite-Link Aware Communication Protocol (S-LACP), is employed over a satellite segment to decrease the residual packet error rate down to an acceptable level without significantly increasing the link delay, as seen by the higher layers. This would allow employing TCP, which is a mature and widely deployed protocol and which has been thoroughly studied and has efficient implementations in several platforms. Furthermore, this approach follows the end-to-end principle in designing the Internet protocols and does not sacrifice the end-to-end semantics, thereby avoiding all the related negative implications on security, fate sharing, end-to-end reliability [16].

In addition to the TCP enhancements, the effect of the S-LACP protocol is validated in a set of tests. This means that the focus of the study is not in evaluating the performance of TCP over satellite links in large but rather in validating the approach of using S-LACP and building up a better overall view of the TCP behavior in the TRANSAT target environment. No S-LACP implementation is in use in tests, but the effect of S-LACP is approximately modeled with tests having lower packet error rate combined with an additional delay due to link-layer retransmissions during an error burst. We show that when S-LACP is used to reduce the residual packet rate as seen by the TCP layer, the performance improves, thereby justifying the approach with the additional layer 2.5 error recovery. Without S-LACP the modeled packet error rate tends to be rather high, not allowing a TCP sender to open the congestion window large enough or later on to retain the congestion window fully open.

We use Network Simulator [13] for modeling the environment and studying the performance of TCP in satellite networks. NS is an object oriented, discrete event-driven network simulator developed at UC Berkeley written in C++ and OTcl, which is very useful for simulating local and wide area networks. Using NS is easy to model, analyse and compare different TCP variations. We choose a Reference TCP implementation, with features assumed to be common and widely deployed in the Internet today. We also use TCP with some improvements, which are intended to improve the performance of TCP over network paths including a satellite link. We call this TCP the Enhanced TCP. The enhancements in use are enhancements described in Standards Tracks of IETF and they are expected to be deployed in a short time.

We first study the behavior of TCP with one connection transferring bulk data. We also study the TCP behavior with several parallel connections. As an additional case, we add competing traffic with TCP to address a more realistic real-world case. We apply an empirical error model to simulate GEO satellite failures in order to cause packet errors. We evaluate and compare the effects of various TCP improvements on TCP behavior under the same network conditions. Finally, we present some concluding remarks, summarizing the outcome of each test case and illuminating the reasons behind the performance results as well as reasons for various kinds of TCP behavior observed.

The rest of the paper is organized as follows. Section 2 describes the test arrangements used in the performance measurements, illustrating the experimentation environment, network configuration and parameters, TCP enhancements and workload models. In Section 3 we present our experiments and performance results. Finally, in Section 4 we present some concluding remarks, summarizing the outcome of each test case and illuminating the reasons behind the performance results as well as reasons for various kinds of TCP behavior observed.

2. TEST ARRANGEMENTS

2.1 S-LACP Protocol

Satellite-Link Aware Communication Protocol (S-LACP) is intended to be employed over a satellite link to enhance transport performance. S-LACP is a satellite-link-specific protocol that logically operates at the link layer but also includes various useful features for mapping IP layer services to the underlying satellite link service. S-LACP uses error control schemes to reduce the packet error rates on the satellite channel. The S-LACP protocol only runs over the satellite link in the forward and return direction. The S-LACP layer shall be considered as layer 2.5 between IP and link layers. In order to decrease the packet error rate (as seen by higher layers) over the satellite, specific error recovery mechanisms are implemented, like ARQ and FEC, for both forward and return direction as well as QoS support mechanisms taking into account the QoS needs of the flows. Flows asking for different QoS will be classified in different queues and will be given different error recovery treatment by S-LACP. ARQ and FEC are selectively applied using several logical S-LACP channels, so that delay sensitive flows are directed over S-LACP channels not performing ARQ (but possibly using FEC).

2.2 Test Environment

The test environment consists of a TCP sender that communicates with another TCP host via a router connected to a satellite link. The Network Simulator is used to simulate the GEO satellite link, routers and the TCP behavior.

In this study, we first ignore the possible congestion problems of the global Internet and focus on the behavior of the Reference TCP and enhanced TCP on the GEO satellite link. Addressing the full problem requires adding a tail network (the Internet), but in the first place, we are interested in the effect of packet losses (over satellite) with different TCP variations. To achieve this we first model a well behaving Internet. Even in this case it is expected that congestion will affect the TCP behavior in some scenarios, as the satellite link is the bottleneck link of the test environment. In order to model the Internet one should run a very large set of tests, as a typical Internet path does not exist (that is why simulating the Internet is very hard). This is left for further study and will be addressed in the performance study of the TRANSAT Phase 2.

We use the network topology showed in Figure 1. The two hosts are connected to two routers, which communicate via a GEO satellite link. The GEO satellite has a bandwidth of 2 Mbits/s and one-way propagation delay of 255 ms. The hosts H1 and H2 are connected to a network of bandwidth 10 Mbits/s. We consider a unidirectional flow of data from host H1 to host H2. Therefore, the workload models we have selected are realized on host H1. The MTU size is taken to be 1000 bytes over each link. The buffer size (the IP router queue) at the Satellite Node 1 is 50 packets. The buffer size at the other nodes is selected large enough to prevent congestion from occurring elsewhere on the end-to-end path.

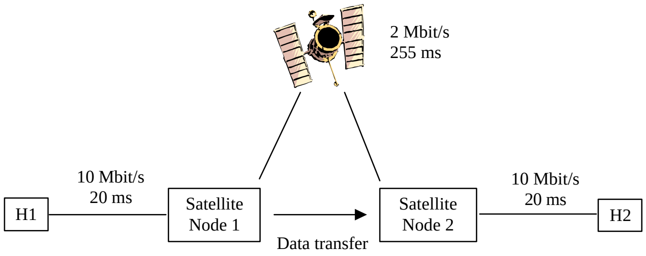


Figure 1. Network Topology simulated to study the effects of the satellite link on TCP

We run tests with different kinds of satellite nodes: satellite nodes without S-LACP and satellite nodes with S-LACP enabled.

In addition to an optimal satellite link with no packet losses as the baseline, we apply an empirical burst error distribution based on actual test data taken from the DSN (Deep Space Network [12]) to cause packet errors. In this model, bursts duration is from 100ms to 500ms with an exponential distribution and the inter-arrival time between two consecutive bursts is from 1 to 30 seconds. The packet error rate (PER) during a burst is 100%. In our test cases, the inter-arrival time is from 1 to 10 seconds. This is because otherwise many shorter test runs may experience no error burst at all whereas our interest is in studying the behavior of TCP versions experiencing at least one error burst.

In order to validate the basic ideas of the S-LACP protocol we use a crude approximation of S-LACP, instead of full implementation of S-LACP in the NS simulator. A rough model on the impact of S-LACP is implemented by reducing the packet-error rate during the error-burst down to 25% and 1%. In addition, we simulate the additional delay due to the S-LACP ARQ mechanism by increasing the delay for the packets affected by the error burst (those not finally dropped). In the first place, the latency of the satellite link is increased by 550ms during bursts errors to model a delay of one additional RTT for retransmitted packets. Additional test cases with higher extra delay up to six times the satellite link latency added to simulate S-LACP retransmissions were run.

Since packets are delivered in sequence by S-LACP, the packets delayed due to S-LACP retransmissions would cause a burst of packets to be sent after the delay. This is a potential problem. We intend to address this in S-LACP final implementation in form of packet pacing to avoid (large) bursts. The details of this are still to be defined in the detailed design of TRANSAT Phase 2 of the project. Since such a packet-pacing function is not available in NS, we do not intend to model the impact of S-LACP correctly by pacing this kind of burst behavior. Instead, a burst of packets is likely to follow the modeled S-LACP delay in our simulations, possibly resulting in somewhat worse performance in some test scenarios than what could be achieved with S-LACP pacing. The results can then be used for validating the need for the S-LACP pacing.

2.3 TCP Variants

We selected a Reference TCP to be used as the baseline against which the performance of the different TCP enhancements is studied.

The Reference TCP is defined with the following features: congestion control (Slow Start, Congestion Avoidance, Fast Retransmit, etc.) as defined in [4], the TCP New Reno [5] algorithm for Fast Recovery, Initial Window of 1 segment, and Delayed ACK threshold was set to 200ms. The TCP SACK option [6] was disabled for the Reference TCP.

The following TCP enhancements were studied:

- Initial window of 4 segments [7]
- SACK option [4] with Conservative SACK algorithm [8,9]
- Limited Transmit [10]
- DAASS – immediate acknowledgement for the first ten segments [11]

In order to show the impact of different TCP enhancements on TCP performance, we compare the Reference TCP to the following TCP variants:

- Reference TCP with an initial window of 4 segments (**Ref, Iwin=4**)
- Ref. TCP with ACK-every-Segment during Slow Start (DAASS) – immediate ack of ten first segments (**Ref, DAASS**)
- Ref. TCP with SACK (**SACK**)
- Ref. TCP with SACK, Limited Transmit (**SACK, LimTran**)
- Enhanced TCP with all enhancements above (**Enh. TCP**)
- Reference TCP with RED (**Ref. + RED**)
- Enhanced TCP with RED (**Enh. TCP + RED**)

The TCP window-scaling option is turned on in all tests as otherwise some TCP transfers would be receiver-window limited due to the large bandwidth-delay product of the end-to-end path. The SACK version used in the tests is the version available in NS, described in [17].

2.4 Workload

We selected two workload models to be used in performance tests: a single unidirectional bulk transfer and four parallel unidirectional bulk transfers. Both workloads under study are tested with and without competing traffic. In single unidirectional bulk transfer the transfer is achieved by using a single TCP connection. The transfer size is 1 MB. In parallel unidirectional bulk transfers we have four TCP connections on the same satellite link in order to study the effect of simultaneous TCP flows and fairness between them. The transfer size is 250 KB for each TCP flow, hence the total transfer size of 4 TCP flows equaling 1MB. We add competing traffic in order to study the implications on the TCP behavior over a congested bottleneck link. A Constant Bit Rate flow of 0.8 Mbit/s up to 1.7 Mbit/s forms the competing traffic.

3. SIMULATION RESULTS

3.1 Optimal case

In this section, we present simulation results for the optimal case in which no burst errors happen. These tests form a baseline for TCP behavior in the target environment with no error-related packet losses. Table 1 shows the elapsed time of one replication for one TCP flow without and with competing traffic. Columns indicate the buffer size (IP router queue) at the Satellite Node, the traffic characteristics and the elapsed time to complete the transfer for different TCP versions (Reference TCP, Reference TCP with initial congestion window of 4 segments (Iwin=4), Ref. TCP with DAASS, Sack TCP, Sack TCP with Limited Transmit and Enhanced TCP).

For reference purposes, the first row of Table 1 shows the optimum case in which the TCP flow does not experience congestion losses. This is achieved by setting the buffer size of Satellite Node 1 up to a very large value (200 packets). This case yields the best possible performance with the given network configuration because neither packet losses nor other factors exists that would sacrifice TCP performance. The Window Scaling option is also turned on in this test case, allowing unlimited utilization of the available link bandwidth.

Table 1 - Elapsed time of one replication for one TCP flow without and with competing traffic

		Elapsed Time (sec)					
Buffer size	**Traffic**	**Ref. TCP**	**Ref., Iwin=4**	**Ref, DAASS**	**SACK**	**SACK, Lim Tran**	**Enhanced TCP**
No competing traffic							
200	1 TCP flow (1MB) (optimum case)	10.69	8.94	9.33	10.69	10.69	8.26
200	1 TCP flow (1MB) Window Scaling OFF	14.62	12.93	13.32	14.62	14.62	12.31
No competing traffic							
50	1 TCP flow (1MB)	60.81	55.89	56.47	15.52	15.51	9.13
50	1 TCP flow (1MB) Window Scaling OFF	14.62	12.93	13.32	14.62	14.62	12.31
With competing traffic							
50	1 TCP flow (1MB) + CBR 0.8 Mbit	63.24	62.4	59.23	12.9	12.92	10.6
50	1 TCP flow (1MB) + CBR 0.8 Mbit Window Scaling OFF	14.69	13.03	13.67	14.69	14.69	12.22
50	1 TCP flow (1MB) + CBR 1.7 Mbit	46.71	42.96	43.43	29.39	29.04	35.06
50	1 TCP flow (1MB) + CBR 1.7 Mbit Window Scaling OFF	37.82	36.53	40.18	30.56	30.00	34.68

The second row shows the effect of not selecting the TCP window-scaling option. This makes the TCP transfer receiver-window limited and does not allow the TCP sender to fully utilize the available bandwidth. This result was used to justify the selection of a window-scaling option for all TCP variants.

Congestion occurs and results in worse performance, when the Satellite Node 1 buffer size is set down to the default value of 50 packets (row 3). This clearly indicates that the satellite link is subject to congestion.

Row four shows the case for the TCP flow with the window scaling option OFF and with default buffer size. Turning the window scaling off effectively prevents the TCP flow from overflowing the router buffer and thereby no congestion occurs. This is because the TCP sender becomes receiver-window limited and is no longer able to increase its congestion window to a large enough value to cause congestion on the Satellite node 1. We do not use this result to justify turning off the window scaling in all tests, because limiting the receiver window in such a simple way does not necessarily help with simultaneous TCP flows and competing traffic where less bandwidth is available per TCP flow and therefore congestion is likely to occur anyway. In addition, it is well known that TCP fast recovery, either with or without SACK, does not work well if the TCP sender is receiver-window limited.

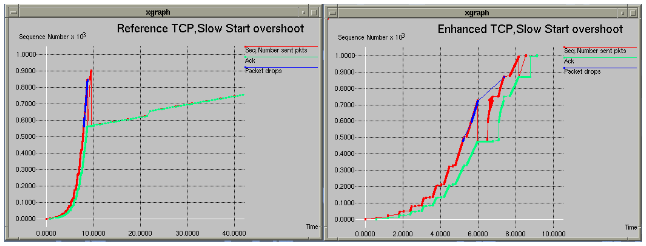


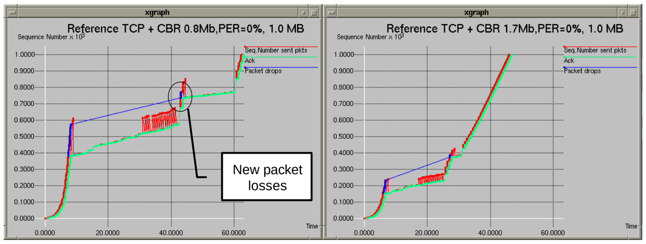
Figure 2. Reference TCP (2a) and Enhanced TCP (2b), Table 1 row 3. Sequence Number and ACK trace when the slow-start overshoot problem occurs.

The selected enhancements either improve or yield similar TCP performance as the Reference TCP, which is an expected result. The elapsed time for SACK TCP, SACK with Limited Transmit and Enhanced TCP (with all the selected enhancements) is much shorter than the elapsed time for Reference TCP in cases where congestion-related losses occur. Reference TCP with larger initial congestion window and Ref. TCP with DAASS yield somewhat better performance in all cases due to faster increase of the congestion window during the initial slow-start phase. When Congestion occurs (rows 3-7), neither the larger initial window nor DAASS are able to improve the performance much, since the slow-start overshoot problem causes a larger number of packet drops at the end of the slow-start phase and it takes longer to recover from such a congestion-related loss burst. In other words, the exponential opening of the congestion window results in overshooting the available bandwidth and loosing multiple segments.

Figure 2 illustrates the slow-start overshoot problem. The Sequence Number of sent packets and the received ACK traces are respectively showed in red and green. Packet drops are indicated in blue. In Figure 2a the TCP sender is in the initial slow start phase, exponentially increasing its congestion window each round trip time. When the router buffer is exhausted, a large number of packets in the window is dropped. This happens because the sender roughly doubles the sending rate each RTT. When the buffer overflows, the congestion window size has reached roughly twice the buffering capacity of the path, resulting in loss of most packets in the latter half of the window. The same phenomenon happens with SACK, but the SACK option allows recovery from such burst loss to be much quicker (see Figure 2b).

Table 1 shows that the Reference TCP performs better with a higher amount of competing traffic than with a lower amount of competing traffic; rows 5 and 7 of Table 1, TCP with a CBR flow of 0.8 Mb/s and with a CBR flow of 1.7 Mb/s, respectively. The reason is that with a higher amount of competing traffic, TCP reacts earlier to congestion and the slow-start overshoot effect is less harmful (less packet losses). Figure 3 illustrates the trace of two such cases. With a 0.8 Mb/s CBR flow (a) new packet losses occur after the first recovery phase (t=45s) because the sudden high number of acknowledged packets allows the TCP sender to transmit a burst of new packets. This could be avoided by tuning the maximum burst of packets that can be sent in response to one acknowledgement. With a 1.7 Mb/s CBR flow, however, this does not happen because less packets are lost during the initial slow start; consequently, after the recovery phase the number of acknowledged packets is less than in the case with 0.8 Mb/s CBR flow and a smaller burst of new packets is sent.

Table 2 shows the elapsed times in one replication for four parallel TCP flows without and with competing traffic. It shows statistics for the fastest and the slowest TCP flow among the four parallel TCP flows. There are two columns more than in Table 1: Reference TCP and Enhanced TCP with RED. The RED Active Queue Management was not used with a single TCP flow case as it was not expected to help due to reasons discussed in section 3.2.



(a) Ref. TCP + CBR 0.8Mb (b) Ref. TCP + CBR 1.7Mb.

Figure 3. TCP behavior with different amounts of competing traffic.

Without competing traffic, congestion occurs during the initial slow start phase and the window scaling option OFF helps to prevent congestion because it avoids overflowing the router buffer. Figure 4 shows the fastest and the slowest flow (red and green line) in the test case without competing traffic (Table 2, row 1,3). Congestion occurs during the initial slow start case; the Enhanced TCP recovers packet losses much quicker than Reference TCP.

With competing traffic, TCP reacts to congestion early and the slow-start overshoot phenomenon is avoided or the effect is less severe as it occurs much earlier. That is why some TCP versions perform better with competing traffic than without. Competing traffic causes the TCP window size to be always below 64KB because TCP reacts to congestion before reaching the 64KB window size; therefore, the Window Scaling option has no effect on the performance in this case (rows 5-6 and 7-8 shows the same results). Table 2 shows that the performance for the fastest and the slowest flow are quite different with each TCP version.

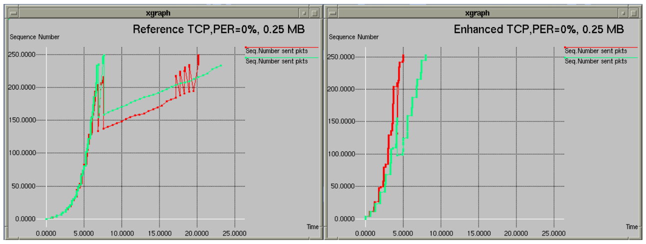
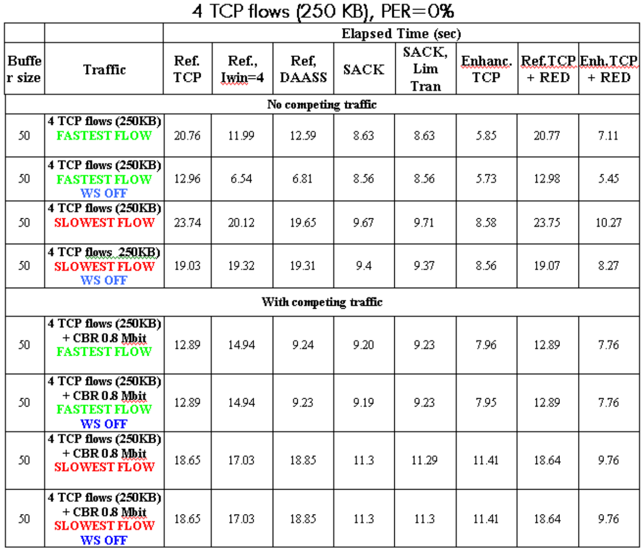


Figure 4. Without competing traffic, congestion occurs during the initial slow start phase. Enhanced TCP recovers packet losses much quicker than Reference TCP. Reference TCP (left side) and Enhanced TCP (right side). 4 TCP flows, FASTEST flow (red line) and SLOWEST flow (green line).

This indicates that in both cases, with and without competing traffic, the available bandwidth is not shared fairly between the four flows. With competing traffic, Active Queue Management is useful in improving the fairness between the TCP flows. The last four rows, indicating four parallel connections with competing traffic, show that Enhanced TCP takes advantage of RED. The fastest flow of Enhanced TCP performs approximately the same with and without RED, but the slowest flow improves its performance, and correspondingly the total performance of four flows, with RED. However, without competing traffic the slow-start overshoot problem remains the main reason for sub-optimal performance and it can not be alleviated with use of RED, because RED is not able to react fast enough to the exponentially increasing queue size during the initial slow-start of the TCP flows. A further analysis of the RED effect is in the next sections of the document.

Table 2. Elapsed time of one replication for four TCP flows without and with competing traffic.



4 TCP flows (250 KB), PER=0%

		Elapsed Time (sec)							
Buffer size	Traffic	Ref. TCP	Ref., Iwin=4	Ref, DAASS	SACK	SACK, Lim Tran	Enhanc. TCP	Ref.TCP + RED	Enh.TCP + RED
No competing traffic									
50	4 TCP flows (250KB) FASTEST FLOW	20.76	11.99	12.59	8.63	8.63	5.85	20.77	7.11
50	4 TCP flows (250KB) FASTEST FLOW WS OFF	12.96	6.54	6.81	8.56	8.56	5.73	12.98	5.45
50	4 TCP flows (250KB) SLOWEST FLOW	23.74	20.12	19.65	9.67	9.71	8.58	23.75	10.27
50	4 TCP flows 250KB) SLOWEST FLOW WS OFF	19.03	19.32	19.31	9.4	9.37	8.56	19.07	8.27
With competing traffic									
50	4 TCP flows (250KB) + CBR 0.8 Mbit FASTEST FLOW	12.89	14.94	9.24	9.20	9.23	7.96	12.89	7.76
50	4 TCP flows (250KB) + CBR 0.8 Mbit FASTEST FLOW WS OFF	12.89	14.94	9.23	9.19	9.23	7.95	12.89	7.76
50	4 TCP flows (250KB) + CBR 0.8 Mbit SLOWEST FLOW	18.65	17.03	18.85	11.3	11.29	11.41	18.64	9.76
50	4 TCP flows (250KB) + CBR 0.8 Mbit SLOWEST FLOW WS OFF	18.65	17.03	18.85	11.3	11.3	11.41	18.64	9.76

3.2 Test cases with burst errors

In this section, we summarize simulation results for a transfer size of 1MB using one TCP Flow without and with competing traffic, and a total transfer size of 1MB using four parallel TCP flows (250 KB for each TCP flow), without and with competing traffic. For brevity reasons, in this paper we just report (see Appendix) the results for the case of one TCP flow (1MB) without competing traffic and four parallel TCP flows (250 KB for each TCP flow), with competing traffic. Table 3, 4 and 5 show statistics of 20 test replications when S-LACP is not used (hence the PER during error bursts is 100%) and when S-LACP is used. In the latter case, PER is reduced to 25% and 1% and packets are delayed during bursts due to S-LACP retransmissions. In the case where there are four flows, statistics of all 80 TCP connections (4 flows each replication x 20 replications) are showed in Tables 6, 9 and 12. Tables also show statistics for the fastest flow and slowest flow for each of the 20 replications ("fastest flow" is defined like the first flow which ends the data transfer among the four concurrent flows). In other words, statistics are calculated selecting the fastest and the slowest flow among the four flows for each replication.

The results show that the selected TCP enhancements improve performance compared to the Reference TCP as could be expected. The median value for elapsed time with each TCP improvement is lower than the median for Reference TCP. The larger Initial congestion window and DAASS help to improve performance during the initial slow start phase. The SACK option helps to recover packet losses more efficiently. Enhanced TCP combining all TCP improvements offers best performance in all cases. The S-LACP allows better performance because of its ability to reduce packet losses, even though it causes additional packet delay.

In the test case with a single transfer size of 1 MB in which S-LACP is used and PER is 1%, the Reference TCP performs the same badly because congestion happens during the initial slow-start (slow start overshoot). This also happens for the TCP SACK and Enhanced TCP, but the SACK option allows packet losses recovery to be very quick (see Figure 2a).

The RED active queue management mechanism is used together with both Reference TCP and Enhanced TCP. RED queue management, however, is not able to help in scenarios without competing traffic.

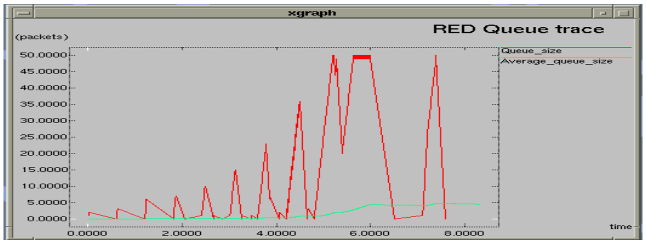


Figure 5. Instantaneus queue size (red line) and average queue size (green line).

Figure 5 shows that the workload used makes the instantaneous RED queue size (red line) very variable and unsteady, in particular during the initial slow-start phase. Therefore, the average queue size (through which RED packet drops are controlled) stays under the minimum threshold for the entire slow-start phase. This is a known phenomenon and RED is known to work best with aggregate traffic.

However, when active queue management is used with competing traffic, the effect of the RED queue management is now visible. With competing traffic and four competing TCP flows, RED helps both improving transfer performance and fairness among different flows. The median value of Table 7 shows that Enhanced TCP takes quite a clear advantage of RED and the Reference TCP is also slightly improved. Table 8 and Table 9 show the fastest flows of the Enhanced TCP performing roughly the same with and without RED, but the slowest flows improve their performance, and correspondingly the total performance of four flows, with RED. In particular, the performance of the Enhanced TCP significantly improved.

When S-LACP is in use, the trend of the improvements with RED is visible with Enhanced TCP but the improvement is not that notable any longer. With Reference TCP the RED queue management is not able to help anymore when the error-related losses become more infrequent. This seems to be due to increasing congestion at the end of the slow-start (the slow-start overshoot problem). The phenomenon remains roughly the same as without competing traffic as the competing traffic is not reacting to congestion but just reduces the amount of available bandwidth for TCP flows.

3.3 Comparing the behavior of different enhancements

In this section, we present some performance graphs to explain the behavior of different TCP versions. Figure 6 allows comparing the initial Slow Start phase of Reference TCP and Enhanced TCP and shows the Sequence Number trace of sent packets (red line) and the received ACK trace (green line). The Enhanced TCP version includes an initial congestion window of four segments and the DAASS option, implemented by immediately acknowledging the first 10 segments. As the figure shows, these improvements allow increasing the congestion window quickly, roughly 5 times faster, even though the high RTT of the satellite link.

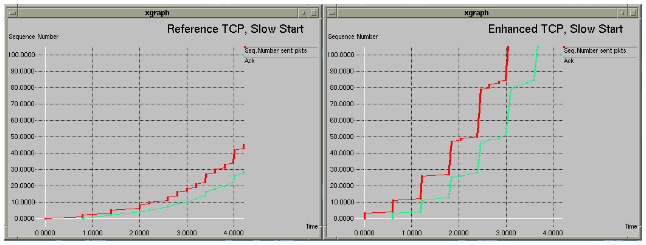


Figure 6. Comparison between the initial Slow Start phase of Reference TCP and Enhanced TCP

Figure 7 shows an example of the SACK ability to quickly recover multiple packet losses after a burst error event occurs. The yellow line shows a burst happening at t = 4.2s and packets 43-48 are lost (blue points). The Reference TCP (Figure 7a), which includes TCP NewReno, is able to recover just one segment per RTT, whereas the Sack option allows a faster recovery because of Selective ACK information (Figure 7b). Figure 8 shows an example of the effect of using S-LACP on the satellite link. Figure 8a indicates the trace of the Reference TCP in the ideal case, no packet losses and no delay. Figure 8b shows the case in which S-LACP is used, causing a residual PER of 1% and an additional delay because of additional retransmissions. In this example, S-LACP is able to recover all packet losses and it only increases the link delay during the burst error period as seen by higher layers. The red line show the sequence number of sent data packets, the green line indicates the sequence number of corresponding data packets when received at the receiver. The blue line shows the arrival of ACK packets at the sender. The burst event (yellow line) is indicated in both figures, even though no packet losses occur in either of the cases. In other words, the yellow line shows where the satellite channel is in the "bad state", causing an additional delay for the S-LACP case.

The ACK trace (blue line) in Figure 8b shows that ACK information arrives later to the TCP sender than for the ideal case shown in Figure 8a (ACKs 42-48 arrive roughly a half second later). The green line, which indicates when packets are received, shows that with S-LACP, the packets 42-48 arrive roughly a half second later than without S-LACP.

Nevertheless, S-LACP significantly improves TCP performance compared to the case in which packets are lost (see Figure 7) because of avoiding unnecessary reductions of the TCP congestion window after packet losses. Even though the S-LACP causes an additional packet delay, no RTO happens. Figure 8b shows that the delay due to S-LACP retransmissions causes a burst of packets (packets 67-78, red line). This phenomenon is not a problem in this example, but could be in other cases. That is why a packet-pacing ability might be useful in S-LACP.

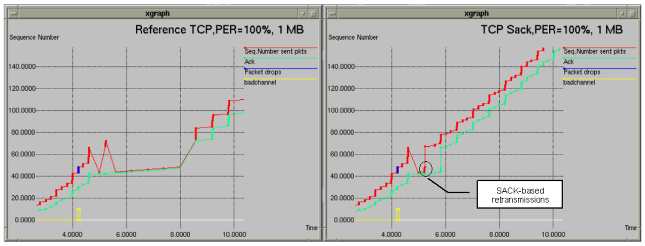


Figure 7. Reference TCP (a) and TCP Sack (b), Sack ability to recover packet losses

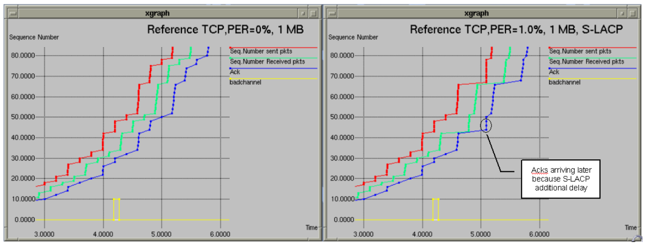


Figure 8. S-LACP effect, (a) ideal case (no errors, no delay), (b) S-LACP case (PER=1% and additional delay)

4. CONCLUSIONS

Several test cases were run to complete the simulation study. In general, the results show that all selected TCP enhancements are helpful compared to Reference TCP. Active queue management is not quite helpful with the level of traffic aggregation used in the different workload models. This is mostly because the major problem with TCP performance in most of the test cases used in this study environment seems to be the slow-start overshoot problem during the initial slow start; the average queue size maintained by the RED active queue management algorithm is not able to react fast enough to the rapid increase of the TCP congestion window during the initial slow start. However, the trend is such that active queue management tends to become more useful when more traffic aggregation is present. We assume that more traffic is aggregated in a typical usage environment for the TRANSAT architecture. In addition, we believe that the use of TCP Control Block Interdependence [14] would be useful in preventing the slow-start overshoot problem from occurring, as it would allow reuse of congestion control parameters from earlier TCP connections, in particular in setting the initial value for the slow-start threshold (sstresh).

When S-LACP is used to reduce the residual packet rate as seen by the TCP layer, the results are encouraging, thereby justifying the approach with the additional layer 2.5 error recovery. Without S-LACP the modeled packet error rate tends to be rather high, not allowing a TCP sender to open the congestion window large enough or later on to retain the congestion window fully open. With S-LACP, the modeled packet error rate remains at a relatively moderate or low level allowing the TCP sender to take advantage of a much larger congestion window.

In addition to the test cases reported in this document, we also ran some test cases with larger S-LACP delay as well as some tests with larger transfer size. In the additional S-LACP tests, the extra delay added to simulate S-LACP retransmissions was increased up to three extra satellite RTTs. Adding three extra RTTs models a case where the retransmission persistency of the S-LACP is tuned so that the S-LACP sender tries to retransmit a lost packet at most three times. The results showed that even adding two extra RTTs does not cause spurious TCP retransmissions timeouts and that TCP performance is comparable with the tests results reported. However, the burst of sent packets following the S-LACP recovery sometimes leads to a different kind of congestion loss patterns and is more likely to be harmful, at least in some test scenarios. This also means that the packet-pacing ability in S-LACP should be considered, in particular if more persistent ARQ is used with S-LACP. When the extra delay was increased to three satellite RTTs, the spurious TCP RTO tends to occur every time an error burst is experienced, resulting in unnecessary TCP retransmissions. The F-RTO [15] algorithm can efficiently prevent unnecessary retransmission and it is expected to be useful in particular if more persistent S-LACP ARQ is in use but also with less persistent ARQ as infrequent delay spikes may still exist.

Additional tests with a transfer size of 10MB were also run with and without S-LACP. Without SLACP, the TCP performance remains very low. The median transfer time is roughly 600s for each TCP version; the selected TCP enhancements yield a slight improvement. This can be explained because frequent error bursts continuously cause new congestion window reductions. Therefore, the congestion window size remains very low for most of the transfer time. In such conditions the SACK extension does not help much. With S-LACP, TCP performance improves significantly as the residual PER is reduced and Enhanced TCP gives the best performance (the median elapsed time being 124.3s). However, the satellite link utilization is still low (roughly one third of the link bandwidth) because the slow-start overshoot phenomenon is still dominating.

5. ACKNOWLEDGMENTS

The authors would like to thank their colleagues in the TRANSAT project for providing helpful comments on earlier versions of the paper. Particularly, we would like to thank A. Heeger, G. Maman and D. Genty from Alcatel Space for their valuable feedback on various aspects of the paper.

6. REFERENCES

[1] Transport Protocol and Communication Resource Management for Mobile Satellite Networks, available at http://telecom.esa.int/telecom/www/object/index.cfm?fobjectid=1030

[2] RFC2488, “Enhancing TCP Over Satellite Channels using Standard Mechanisms”, M. Allman, D. Glover, L. Sanchez

[3] “Research Issue for Transport Protocols in Satellite IP Networks”, I. Akyildiz, G. Morabito, S. Palazzo, IEEE Personal Communications, June 2001

[4] RFC2581, “TCP Congestion Control”. M. Allman, V. Paxson, W. Stevens. April 1999.

[5] RFC2582, “The NewReno Modification to TCP's Fast Recovery Algorithm”. S. Floyd, T. Henderson. 1999.

[6] RFC2018, “TCP Selective Acknowledgement Options”. M. Mathis, J. Mahdavi, S. Floyd. Oct 1996.

[7] RFC2290, “Increasing TCP's Initial Window”. M. Allman, S. Floyd, C. Partridge. October 2002.

[8] RFC3517, “A Conservative Selective Acknowledgment (SACK)-based Loss Recovery Algorithm for TCP”. E. Blanton, M. Allman, K. Fall, L. Wang. April 2003.

[9] RFC2883, “An Extension to the Selective Acknowledgement (SACK) Option for TCP”. S. Floyd, J. Mahdavi, M. Mathis, M. Podolsky. July 2000.

[10] RFC 3042, “Enhancing TCP's Loss Recovery Using Limited Transmit”. M. Allman, H. Balakrishnan, S. Floyd. January 2001.

[11] RFC2760, “Ongoing TCP Research Related to Satellites”. M. Allman, Ed., S. Dawkins, Feb 2000.

[12] J. Murphy, “Measuring & Modelling A Satellite Link”, http://www.eeng.dcu.ie/~murphyj/the/the/node60.html

[13] Network Simulator, http://www.isi.edu/nsnam/ns/

[14] RFC 2140, “TCP Control Block Interdependence”. J. Touch. April 1997.

[15] “F-RTO: A New Recovery Algorithm for TCP Retransmission Timeouts”. P. Sarolahti, M. Kojo, and K. Raatikainen. Helsinki University, Dep of Computer Science, Technical Report C-2002-07, Feb 2002.

[16] RFC 3135, “Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations” – J. Border, M. Kojo, J. Griner, G. Montenegro, Z. Shelby

[17] “Comparisons of Tahoe, Reno, and Sack TCP”. Fall, K., and Floyd, S. December 1995”.

7. APPENDIX

In this appendix, we report the results for the test cases of one TCP flow (1MB) without competing traffic and four parallel TCP flows (250 KB for each TCP flow), with competing traffic.

Transfer size 1MB, one TCP flow without competing traffic

Table 3. Statistics of 20 test replications when S-LACP is not used.
The PER during error bursts is 100%

		Ref. TCP	Ref., lwin=4	Ref, DAASS	SACK	SACK, Lim	Enh. TCP	Ref. + RED	Enh. + RED
Elapsed Time (sec)	Min	26.07	14.09	14.50	14.94	14.93	9.13	23.10	8.88
	Max	113.92	119.80	119.76	134.88	126.29	84.40	113.92	84.40
	25% Perc.	53.27	24.77	22.32	18.60	18.36	9.98	53.96	9.49
	Median	65.92	42.62	38.39	53.24	57.46	13.65	65.92	14.70
	75% Perc.	82.93	70.40	66.84	83.73	79.56	27.34	82.93	27.48
	Average	68.10	49.41	47.84	58.40	54.44	25.37	67.76	25.77
Avg. Throughput (KB/s)		17.31	28.66	31.04	28.40	29.78	67.47	17.61	66.80

Table 4. Statistics of 20 test replications when S-LACP is used.
The residual PER during error bursts is 25%

		Ref. TCP	Ref., lwin=4	Ref, DAASS	SACK	SACK, Lim	Enh. TCP	Ref. + RED	Enh. + RED
Elapsed Time (sec)	Min	21.35	21.62	14.12	11.63	11.63	8.91	21.35	8.88
	Max	64.81	99.84	85.14	55.36	56.55	63.41	121.29	63.41
	25% Perc.	45.32	28.94	31.93	15.05	14.85	9.29	32.59	8.90
	Median	56.28	44.12	49.90	15.94	15.98	9.73	52.10	9.75
	75% Perc.	61.50	56.63	57.63	23.81	23.22	12.99	61.98	13.13
	Average	50.87	46.53	46.22	21.78	21.43	14.11	51.07	14.11
Avg. Throughput (KB/s)		22.07	26.43	25.90	56.13	56.78	89.16	23.15	89.57

Table 5. Statistics of 20 test replications when S-LACP is used.
The residual PER during error bursts is 1%

		Ref. TCP	Ref., lwin=4	Ref, DAASS	SACK	SACK, Lim	Enh. TCP	Ref. + RED	Enh. + RED
Elapsed Time (sec)	Min	16.04	12.99	11.79	11.60	11.60	8.61	16.91	8.88
	Max	124.49	95.33	77.62	21.85	21.85	19.52	124.50	19.52
	25% Perc.	46.53	33.03	41.45	13.64	13.33	9.31	58.52	8.93
	Median	61.11	56.17	57.01	15.81	15.80	9.69	62.49	9.58
	75% Perc.	62.05	57.18	57.92	16.25	16.13	11.82	64.48	11.78
	Average	54.39	49.04	48.96	15.46	15.33	10.80	62.50	10.88
Avg. Throughput (KB/s)		24.34	27.66	27.08	66.43	67.13	96.13	20.40	96.37

Transfer size 4x250KB, four TCP flows with competing traffic

Table 6. Statistics of all 80 connections (4 flows each replication x 20 replications).
S-LACP is not used. The PER during error bursts is 100%

		Ref. TCP	Ref., lwin=4	Ref, DAASS	SACK	SACK, Lim	Enh. TCP	Ref. + RED	Enh. + RED
Elapsed Time (sec)	Min	7.57	6.01	6.50	7.57	7.54	4.99	7.57	4.99
	Max	64.68	108.24	36.91	68.72	44.18	40.96	45.55	40.96
	25% Perc.	17.70	16.52	14.71	12.31	12.44	9.64	18.10	9.25
	Median	22.09	19.07	18.49	17.58	17.62	12.15	21.80	11.09
	75% Perc.	30.49	23.31	22.58	28.54	29.90	16.65	28.54	15.30
	Average	24.24	21.28	19.30	21.27	20.65	14.19	23.85	12.75
Avg. Throughput (KB/s)		11.82	13.74	15.06	14.93	15.10	21.17	11.88	23.09

Table 7. Statistics of the fastest flow for each of the 20 replications.
S-LACP is not used. The PER during error bursts is 100%

		Ref. TCP	Ref., lwin=4	Ref, DAASS	SACK	SACK, Lim	Enh. TCP	Ref. + RED	Enh. + RED
Elapsed Time (sec)	Min	7.57	6.01	6.50	7.57	7.54	4.99	7.57	4.99
	Max	33.44	22.19	18.72	35.52	33.61	11.68	33.44	10.44
	25% Perc.	14.15	13.73	9.24	10.30	10.30	7.37	14.15	7.03
	Median	16.49	14.94	11.19	13.41	12.90	7.97	16.11	7.76
	75% Perc.	18.99	16.23	14.25	18.43	18.33	9.26	19.22	9.06
	Average	17.95	14.23	12.01	15.44	15.42	8.23	17.94	7.84
Avg. Throughput (KB/s)		15.59	19.55	22.51	19.29	19.27	32.27	15.61	33.45

Table 8. Statistics of the slowest flow for each of the 20 replications.
S-LACP is not used. The PER during error bursts is 100%

		Ref. TCP	Ref., lwin=4	Ref, DAASS	SACK	SACK, Lim	Enh. TCP	Ref. + RED	Enh. + RED
Elapsed Time (sec)	Min	18.49	16.93	18.76	12.23	12.29	11.61	18.65	10.03
	Max	64.68	108.24	36.91	68.72	44.18	40.96	45.55	40.96
	25% Perc.	24.07	22.88	21.41	18.56	16.87	15.37	23.45	13.77
	Median	32.34	26.78	25.93	29.05	26.20	22.40	29.58	15.62
	75% Perc.	37.85	28.90	30.82	36.14	34.12	24.74	37.85	20.81
	Average	32.60	30.94	26.14	29.39	26.05	21.13	30.90	18.45
Avg. Throughput (KB/s)		8.46	9.36	9.99	10.59	11.30	13.15	8.75	15.23

Table 9. Statistics of all 80 connections (4 flows each replication x 20 replications).
S-LACP is used. The residual PER during error bursts is 25%

		Ref. TCP	Ref., lwin=4	Ref, DAASS	SACK	SACK, Lim	Enh. TCP	Ref. + RED	Enh. + RED
Elapsed Time (sec)	Min	7.48	8.19	6.50	7.50	7.45	5.27	7.48	5.27
	Max	32.91	25.82	29.03	31.96	31.55	34.16	30.75	26.50
	25% Perc.	17.56	15.46	11.77	10.56	10.69	9.29	17.53	8.40
	Median	19.42	16.82	16.85	12.86	13.29	10.29	19.53	9.64
	75% Perc.	21.85	18.64	19.19	19.96	21.31	11.61	22.57	10.97
	Average	19.72	17.16	16.36	15.46	15.65	10.98	19.93	10.44
Avg. Throughput (KB/s)		13.77	15.15	16.84	18.79	18.52	24.65	13.63	25.96

Table 10. Statistics of the fastest flow for each of the 20 replications
S-LACP is used. The residual PER during error bursts is 25%

		Ref. TCP	Ref., lwin=4	Ref, DAASS	SACK	SACK, Lim	Enh. TCP	Ref. + RED	Enh. + RED
Elapsed Time (sec)	Min	7.48	8.19	6.50	7.50	7.45	5.27	7.48	5.27
	Max	22.69	16.61	16.87	22.67	22.73	10.33	22.69	9.36
	25% Perc.	12.29	13.51	9.33	9.27	9.24	7.70	12.71	7.76
	Median	15.29	15.07	10.68	9.68	9.89	8.02	14.68	7.89
	75% Perc.	17.84	15.37	12.99	12.81	12.55	9.30	17.30	8.46
	Average	14.83	14.10	11.39	10.94	11.02	8.13	14.59	7.86
Avg. Throughput (KB/s)		18.55	18.37	23.22	24.41	24.23	31.59	18.74	32.44

Table 11. Statistics of the slowest flow for each of the 20 replications
S-LACP is used. The residual PER during error bursts is 25%

		Ref. TCP	Ref., lwin=4	Ref, DAASS	SACK	SACK, Lim	Enh. TCP	Ref. + RED	Enh. + RED
Elapsed Time (sec)	Min	15.26	15.55	16.84	11.45	11.41	10.18	16.39	9.86
	Max	32.91	25.82	29.03	31.96	31.55	34.16	30.75	26.50
	25% Perc.	20.59	18.79	19.10	12.13	13.49	11.60	20.64	10.20
	Median	21.63	20.26	21.30	20.96	18.37	12.93	23.59	12.01
	75% Perc.	26.97	22.54	23.37	26.35	28.87	15.39	28.57	14.15
	Average	23.71	20.61	21.51	20.16	20.36	14.96	24.20	14.14
Avg. Throughput (KB/s)		10.92	12.32	11.84	14.36	14.10	18.27	10.67	19.60

Table 12. Statistics of all 80 connections (4 flows each replication x 20 replications).
S-LACP is used. The residual PER during error bursts is 1%

		Ref. TCP	Ref., lwin=4	Ref, DAASS	SACK	SACK, Lim	Enh. TCP	Ref. + RED	Enh. + RED
Elapsed Time (sec)	Min	7.98	10.54	9.24	8.08	8.06	6.75	7.95	6.77
	Max	26.29	30.32	27.85	26.79	18.66	19.20	26.29	20.00
	25% Perc.	16.19	15.38	11.29	10.10	10.14	8.69	16.48	8.09
	Median	18.11	16.76	17.46	10.99	11.22	9.43	18.41	9.28
	75% Perc.	19.16	17.47	18.91	12.00	11.97	11.40	19.04	10.04
	Average	17.63	16.83	15.56	11.67	11.61	10.01	17.83	9.56
Avg. Throughput (KB/s)		14.70	15.27	17.49	22.25	22.16	25.92	14.52	27.17

Table 13. Statistics of the fastest flow for each of the 20 replications.
S-LACP is used. The residual PER during error bursts is 1%

		Ref. TCP	Ref., lwin=4	Ref, DAASS	SACK	SACK, Lim	Enh. TCP	Ref. + RED	Enh. + RED
Elapsed Time (sec)	Min	7.98	10.54	9.24	8.08	8.06	6.75	7.95	6.77
	Max	19.66	17.30	17.47	12.19	11.89	8.86	20.06	10.44
	25% Perc.	13.10	14.53	9.32	9.35	9.36	7.79	13.10	7.68
	Median	13.71	15.13	9.44	9.64	9.57	8.05	14.45	7.83
	75% Perc.	15.41	15.40	11.76	9.97	10.04	8.25	15.95	8.00
	Average	13.98	14.60	11.06	9.81	9.77	7.95	14.23	7.92
Avg. Throughput (KB/s)		18.58	17.36	23.64	25.68	25.78	31.60	18.32	31.80

Table 14. Statistics of the slowest flow for each of the 20 replications.
S-LACP is used. The residual PER during error bursts is 1%

		Ref. TCP	Ref., lwin=4	Ref, DAASS	SACK	SACK, Lim	Enh. TCP	Ref. + RED	Enh. + RED
Elapsed Time (sec)	Min	16.85	12.29	17.62	11.11	11.28	9.38	16.61	9.89
	Max	26.29	30.32	27.85	26.79	18.66	19.20	26.29	20.00
	25% Perc.	19.00	17.33	18.95	11.51	11.54	11.51	18.99	10.02
	Median	19.42	18.21	19.19	11.95	13.32	11.63	19.42	10.36
	75% Perc.	21.27	19.92	19.68	14.95	15.68	11.97	21.30	12.40
	Average	20.38	19.23	19.80	14.26	13.84	12.32	20.40	11.86
Avg. Throughput (KB/s)		12.41	13.46	12.74	18.60	18.61	20.90	12.39	22.00