Data Acceleration and Reduction Technology

Thava Iyer, Roksana Boreli, Golam Sarwar, Christoph Dwertmann

***Abstract*—We present a new rate based congestion control and scheduling mechanism, DART, which has been implemented in the 7-ip satellite gateway products. DART is based on SCPS-TP framework and includes integrated data compression. We evaluate the performance of this scheme on IPSTAR and Inmarsat BGAN satellite systems against a number of TCP flavors previously proposed for use on satellite links. We show that on congested satellite links the proposed scheme outperforms the established techniques. We additionally highlight the benefits of compression when fully integrated into the transport protocol. Finally, we demonstrate fairness of the proposed scheme to other flows.**

I. INTRODUCTION

Since the mid 80's, there has been a large amount of effort in developing both new satellite technologies and systems and transport protocols for satellite links. It has been shown [1] that today's most commonly used transport protocol, TCP, is not well suited to long delay links. The inherent problems that the Adaptive Increase Multiplicative Decrease (AIMD) window based congestion control scheme used in TCP has on long delay links is related to the maximum window size, slow start phase and various time-outs. Improvements to TCP have resulted in a number of RFCs addressing this issue [1], nevertheless there is space for further improvements.

The on-going development and evolution of new satellite technologies brings new challenges to the performance of different transport protocols or versions of TCP. Traditional satellite systems were designed to have bandwidth dedicated to individual links rather than shared bandwidth as is experienced on the public Internet. This simplifies the transport protocol requirements as it removes the need to deal with congestion. A combination of one way downlink satellite service with higher bandwidth and higher delay and a terrestrial (e.g. dial-up) link with a significantly lower bandwidth and lower delay have introduced an imbalance problem, for which specific transport protocols have been designed so that they outperform others. This kind of imbalance may also be experienced in satellite systems designed exclusively for deep space missions, where the majority of e.g. space research data or images is downloaded and the upload bandwidth requirements are minimal. Finally, the Low Earth Orbit (LEO) satellite systems and the emerging store and forward systems used for more recent deep space missions [2] introduce an additional problem of having periods where a link may not be available, which can have a different optimum solution. To further complicate the ultimate answer to what is the best reliable transport protocol for satellite links, there has been an on-going disparity between the the commercial products, which are in the majority based on proxy solutions [1] and research work which quite rightly argues against proxies as these break the end to end TCP connection.

Despite the maturity of the subject, only a limited number of publications present results of experimental performance on satellite links. Additionally, more recent publications [2], [3] do not consider a very well established Space Communications Protocol Suite transport protocol (SCPS-TP) [4].

Our primary interest is in the performance of reliable transport protocols on satellite systems which are currently used for civilian applications like Internet access or business data services. Such systems are common in large countries like Australia, US or Canada and also in countries with a rapidly growing infrastructure like India or China. There has been a number of new satellite network deployments in recent times [5],[6], [7]. We consider IPSTAR to be a good representative of an Internet access service for remote areas, as it is widely used in a number of Asian countries and in Australia to provide government subsidised rural data services. Similarly, Inmarsat BGAN represents the major global mobile satellite service used for business data services. Both systems share a common bandwidth between a number of users, thereby adding the issue of congestion to the long delay experienced by all satellite services.

Although wide spread, satellite services continue to have lower bandwidth and to be more highly priced compared to both fixed and wireless broadband services in urban areas. Therefore, data compression has also been a relevant technology as it enables more efficient bandwidth use. Some industry solutions include compression, however due to the proprietary nature of implemented technology there has been limited published material on the performance of combined transport protocol and compression systems [8].

In this paper, we present a new congestion control and scheduling mechanism for use in a transport protocol designed for shared bandwidth satellite services. Additionally, we present a method to integrate data compression into the transport layer, which provides potential for further improvements in the use of satellite bandwidth. The new mechanisms has been implemented within the SCPS-TP protocol reference implementation [9] and the proposed scheme is used in the 7-ip satellite gateway router product [10]. We present results of the experimental evaluation of the performance of this scheme, with and without compression, and compare it to the performance of selected TCP variants under similar network conditions. The schemes being evaluated include both proxy and end to end solutions. To the best of our knowledge, this is the first publication presenting any experimental results on BGAN and the first evaluation of TCP variants on IPSTAR.

We analyse the results and provide additional insights into the performance of various schemes, and highlight the impact of compression. We demonstrate that the proposed mechanisms results in a significantly improved bandwidth use on live satellite services while being fair to other flows.

The rest of the paper is structured as follows. Section II provides an overview of the related work. Section III presents a description of the proposed congestion control and compression mechanisms. The following section presents the experimental setup for live satellite tests, followed by the results in analysis presented in Section V. In Section VI we evaluate fairness to other flows. Section VII presents conclusions and a discussion of future work.

II. RELIABLE TRANSPORT PROTOCOLS FOR SATELLITE

TCP versions most commonly used in satellite communications are described below. We only present non-proprietary solutions with known algorithms.

The main goal of having a satellite specific version of TCP, or a performance enhancing proxy, is to be able to fully utilise the bandwidth on the satellite link and to handle congestion in a responsive way. The additional goal is fairness to other flows sharing the same link. For end-to-end approaches, fairness to standard TCP Reno is also required.

With the most commonly used TCP version, Reno, a large window size, selective acknowledgment (SACK) and the Selective Negative Acknowledgment (SNACK) options are beneficial to improve performance on long delay links [1]. For the window based congestion control, a larger window size close to the bandwidth delay product enables higher data rates. However, this does not improve the responsiveness to congestion events, as the receiver feedback will arrive later (due to increased delay) than in the terrestrial links.

TCP Hybla [3] has been specifically designed for long delay links. It implements an enhancement of the standard TCP congestion control algorithms for both the slow start and the congestion avoidance phases. The aim of the Hybla congestion control is to provide connections on long delay links the same growth of the congestion window as what a reference lower delay connection would have. To achieve this, it estimates channel bandwidth, uses time stamps to measure the round-trip time (RTT) and additionally uses SACK.

TCP Vegas also includes new mechanisms for congestion detection during the slow start and congestion avoidance. These are based on measurements of RTT and congestion estimates based on the variations of the RTTs for individual packets. The basic assumption is that increasing RTT for individual packets can be interpreted as an indicator of increasing congestion on the link. The performance gain compared to other variants of TCP and Reno continues to be a subject for debate. As Vegas is used in selected commercial products, we have included it in this evaluation.

The Space Communications Protocol Suite (SCPS) and the satellite communications specific transport protocol SCPS-TP have been developed by the Consultative Committee on Space Data Systems (CCSDS) for use on long delay links [4]. SCPS-TP is a well established protocol and has been used in a number of products [11]. SCPS-TP is implemented [9] as a TCP proxy, which includes three options for congestion control: Vegas, Van-Jacobsen which uses the SNACK option and the pure rate control which does not use any congestion control algorithm.

Our early experimental work with SCPS-TP indicated limited performance gains on congested satellite links (as presented in Section V), therefore we decided to develop a new congestion control mechanism within the SCPS-TP framework. Additionally, our goal was to fully integrate data compression within the same framework in order to improve bandwidth use on the satellite links. The following section outlines the new mechanisms.

III. DATA ACCELERATION AND REDUCTION TECHNOLOGY (DART)

DART is comprised of the compression engine, a rate control algorithm and the scheduling algorithm, which are all integrated within SCPS-TP.

The SCPS-TP gateway has two logical sides: the LAN side which uses TCP and the long delay link side which uses SCPS-TP. It is important to note that SCPS-TP gateway works within a route scope, which relates to a specific satellite link. All transport protocol (TCP or SCPS-TP) sockets belonging to this route are considered as a group. Data from TCP sockets on the LAN side is first buffered, then compressed, buffered again on the SCPS-TP side and transmitted over the link. The scheduling algorithm determines how the associated transport protocol sockets are served and the satellite specific congestion control regulates the transmitted rate.

A. Compression Engine

The compression engine uses Lempel-Ziv Markov chain algorithm (LZMA) [12] lossless compression. It includes a method to check for non-compressible data, i.e. the compression engine will not increase the amount of data compared to the original data payload. A number of TCP segments may be compressed jointly, to maximise the compression efficiency. The resulting compressed data in the SCPS-TP side buffer is therefore ready to be sent over the satellite link.

When a data packet is about to be transmitted, the DART mechanism checks both the congestion control variables and the scheduling algorithm conditions and allows the packet to be transmitted only if all of those are met.

B. Congestion Control Algorithm

The congestion control algorithm developed for SCPS-TP is based on the concept of a virtual buffer, which contains all data in flight, i.e. the data which has been transmitted by the sender but is yet to be acknowledged by the receiver. Periodically, after a specific control time period, the sender calculates the target data rate and the optimal length of the virtual buffer. The calculation is based on the received data rate, fullness of the virtual buffer and a number of parameters. In line with SCPS-TP, the algorithm works within a route scope and all transport layer connections are jointly considered.

Figure 1 shows the logic of the congestion control mechanism. The actual data length in the virtual buffer l_i is monitored, on the sender side, at the end of every $i-th$ control interval. This is done by observing the acknowledged data packets. There are four congestion control operating regions, which are determined by three threshold parameter values and the maximum allowable length of the virtual buffer B_{max}. The ramp-up region starts at the beginning of data transfer and ends when l_i reaches the threshold γ; the core control region is delimited by β; the saturated region is delimited by α; and the guard band region by B_{max}. B_{max} is determined by the combination of delay and maximum bandwidth on the link.

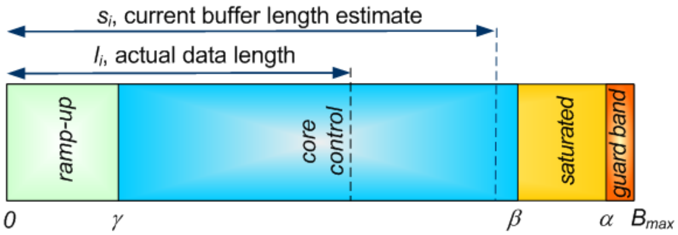

Figure 1. Virtual buffer parameters used in the congestion control mechanism

At the end of every $i-th$ control interval of duration T_c, the amount of data received and the received rate R_i are calculated from the receiver feedback. The average received rate in the last N control intervals $\overline{R}_i^N$ is then calculated as a moving average :

$$\overline{R}_i^N = \frac{\sum_{j=0}^{N-1} R_{i-j}}{N} \tag{1}$$

The value of $\overline{R}_i^N$ is used to calculate the target virtual buffer size and the target data rate for the following, $(i+1)-th$ control period. The operating region determines the method used in the calculations.

1) Ramp-up Region: The initial region for rate and virtual buffer length growth is similar to the slow start in TCP, i.e. it is designed for aggressive growth.

The target virtual buffer size for the $(i+1)-th$ control period s_{i+1}, is calculated as:

$$s_{i+1} = l_i + (s_i - l_{i-1}) + \delta \tag{2}$$

Where:

s_i is the target buffer size for the $i-th$ control period (calculated at the end of $(i-1)-th$ period).

l_i is the length of the actual data in the virtual buffer at the end of i-th control period.

δ is the growth parameter, i.e. increment of data for the control period (in bytes).

s_{i+1} is capped by the parameter α.

$$s_{i+1} = min(s_{i+1}, \alpha) \tag{3}$$

The current target rate r_{i+1} for the $(i+1)-th$ control period is calculated as:

$$r_{i+1} = r_i + \delta/T_c \tag{4}$$

Where T_c is the duration of the control interval.
r_{i+1} is limited to the maximum rate r_{max}.

$$r_{i+1} = min(r_{i+1}, r_{max}) \tag{5}$$

2) Core Control Region: Within the core control region, the aim is to have aggressive rate control and maximum responsiveness to congestion events.

To calculate the target rate and the virtual buffer size, we use both the the short and long term average received rates. We estimate the size or the target virtual buffer for the $(i+1)-th$ control period as:

$$s_{i+1} = (1 + (1-\eta)^2) \cdot (\overline{R}_i^N / r_{max}) \cdot B_{max} \tag{6}$$

Where $\overline{R}_i^N$ is the average received rate for the last N control periods, and

$$\eta = \begin{cases} \frac{\overline{R}_i^N}{r_{max}} - \lambda, & \text{if } \frac{\overline{R}_i^N}{r_{max}} \geq \lambda \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

λ is the range factor parameter.

The current target rate for the $(i+1)-th$ control period is calculated as:

$$r_{i+1} = \overline{R}_i^2 + (1 - (\overline{R}_i^2/r_{max})^2] \cdot \delta/T_c \tag{8}$$

Where $\overline{R}_i^2$ is the average received rate for the last $N = 2$ control periods. The rate is limited to r_{max} by equation (5).

3) Saturation Region: This region is characterised by less aggressive rate growth.

The target buffer size for the next control period s_{i+1}, is calculated by:

$$s_{i+1} = l_i + s_i - l_{i-1}) + \phi \tag{9}$$

Where:

$$\phi = minR_{i-1} \cdot T_c, (\overline{R}_i^N / r_{max}) \cdot \delta \cdot (\beta - l_i)/(\beta - \alpha) \tag{10}$$

α and β are the virtual buffer threshold parameters, as per Figure 1.

The current target rate for the $(i+1)$-th control period is set to the maximum rate, $r_{i+1} = r_{max}$.

4) Guard Band Region: The target buffer size for the next control period s_{i+1}, is equal to α. The current target rate for the next control period is the same as in the saturation region.

C. Scheduling Algorithm

The scheduling algorithm is used to make a decision whether to allow a packet to be transmitted over the satellite link. All SCPS-TP sockets from the socket list are served one at a time. For a socket to be allowed to transmit a packet, a combination of scheduling and congestion control mechanism conditions needs to be satisfied. If those are not met, the scheduling mechanism will move to the next serving socket and the current socket will be have the next opportunity in the following round.

The scheduling algorithm includes the following main conditions which are applied in the given order.

1. If the current socket has a packet to be retransmitted, then allow the packet based on the current target rate regardless of the sockets it belongs to. I.e. give priority to retransmission packets.

2. If the last transmitted data packet is from the same socket, then do not allow the data packet to be sent; allow in the subsequent attempt. This condition enables fair sharing of the connection between all B side sockets in a route.

3. Ensure that the virtual buffer size and current target rate conditions described in this section are satisfied, based on the region of congestion control. In order to avoid bursty traffic within a control period, the period is divided into a number of burst intervals within which the rate conditions are applied.

In the following section we present details of our experimental setup used to evaluate the performance.

IV. EXPERIMENTAL SETUP

All our experiments were conducted in the NICTA Australia Technology Park Laboratory in Sydney. Our experimental setup is presented in Figure 2. We use two satellite services which both have a similar configuration, providing an IP connection to the public Internet. Data is being transmitted from the application PCs, through the router which includes the selected transport protocol, via the satellite modem, the two way satellite link and the Satellite station to our test server.

We use a dual approach: for the SCPS-TP and DART, we use the SCPS-TP gateway on both sides, which is a proxy and breaks up the end to end TCP connection. We use the SCPS-TP reference implementation [9], which was heavily modified to include DART algorithms described in Section III. We have also implemented some performance improvements, to lower the processing power required for SCPS-TP and to enable higher data rate handling. All other TCP flavors use the end to end approach. For those, we use the Linux kernel implemented TCP variants and the gateway router in Figure 2 just routes traffic and captures experimental data. In order to avoid the built-in IPSTAR TCP proxy, we use IP tunneling.

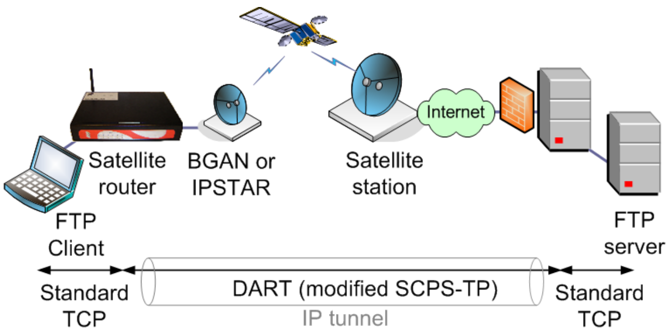

Figure 2. Experimental setup for live tests

We use the Linux Wget application [13] as an FTP client, with a number of test files which have non-compressible content: 10MB file for IPSTAR experiments and a 3MB file for BGAN experiments. To test compression, we use the Canterbury corpus files [14], which are a common reference for evaluating the performance of lossless compression.

A. *Characteristics of Satellite Links*

In previous work [15] we have characterised the IPSTAR satellite network. IPSTAR uses shared access over radio channels by dividing the available bandwidth (6Mbit/sec downlink and 4Mbit/sec uplink) into service plans. The plans are implemented by a combination of oversubscription on each satellite channel and shaping at the Internet Point of presence (POP). The satellite service can have both congestion and errors on the link, although the congestion experienced in long term experiments is low. The network RTT characterised during our long term experiments show, for large packet sizes, an average RTT of greater than 1sec. Published results indicate an operating bit error rate (BER) of 10^{-7} [16].

Inmarsat BGAN also uses shared bearers, with a nominal data rate of 492kbit/sec on both downlink and uplink [5]. Access control is implemented using TDM/TDMA [17]. This creates a coarse granularity in available data rates when a number of users share the link, which can be heavily congested. The network RTT and loss characterised during our experiments show an average RTT of greater than 1.3sec and very occasional packet errors. This is consistent with [18] which defines BER lower than IPSTAR.

Other satellite networks of interest, e.g. DVB-RCS [18], would have similar or lower error rates, so congestion can be seen as the main issue on these links. Additionally, we may experience congestion on the Internet, between the satellite gateway and NICTA server.

The following section presents a summary of tests on IPSTAR and BGAN performed for the SCPS-TP pure rate control transport protocol, DART and different TCP versions.

V. EXPERIMENTAL RESULTS AND ANALYSIS

We perform a number of experiments over the IPSTAR and Inmarsat BGAN satellite networks, at different times of the day. Groups of experiments were performed at the same time for all transport protocol options considered, to minimise the impact of satellite network load conditions on test results.

We measure the goodput as the average data rate received by the application. This takes into account different overhead which the SCPS-TP or the TCP variants may have, e.g. time stamps. We also observe the number of retransmissions for all the experiments. The average goodput and the standard deviation of goodput are presented in the following figures.

A summary of the download results from ten IPSTAR experiments is presented in Figure 3, for a single data connection using DART, SCPS-TP and selected TCP variants. Additional experiments were performed with Vegas congestion control included in the SCPS-TP reference implementation, however the results are very similar to the Linux kernel Vegas results.

Figure 4 presents experimental results for the same protocols with a competing UDP traffic flow of 300-500kbit/sec. We use a non-compressible 10MB file for all experiments presented in figures 3 and 4.

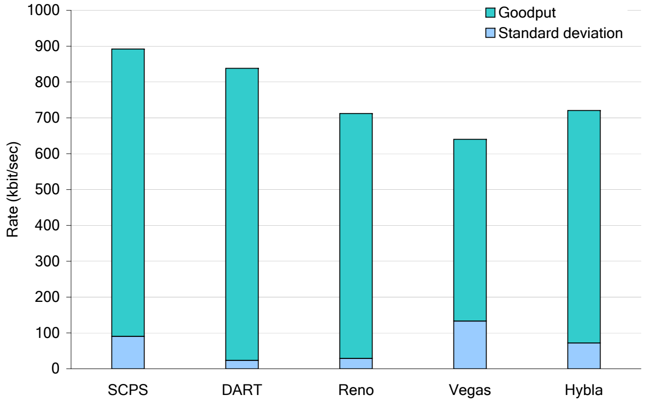

Figure 3. Average and standard deviation of goodput for 10 IPSTAR experiments, for a single connection using DART, SCPS-TP and TCP variants

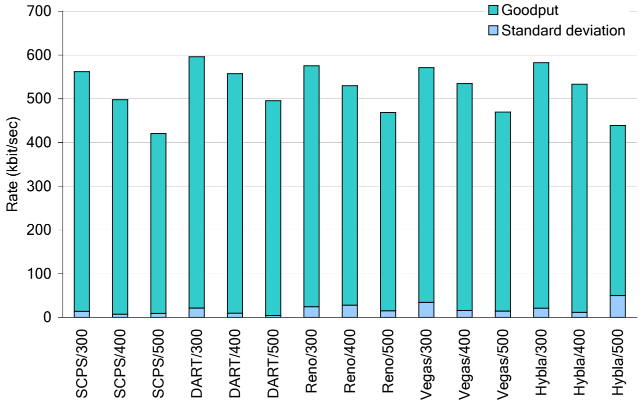

Figure 4. Average and standard deviation of goodput for 10 IPSTAR experiments, for a single connection using DART, SCPS-TP and TCP variants and with a competing UDP flow of 300, 400 and 500kbit/sec

It can be observed that SCPS-TP performs well on an uncongested link. However, as the pure rate algorithm does not have any congestion control, the performance degrades with increasing congestion on the link. This can also be observed by the increasing number of retransmissions, not shown here due to space constraints. DART is the second best and Hybla also performs well compared to other TCP variants, consistent with [3].

Results from Inmarsat BGAN experiments are presented in Figure 5. We show goodput and the standard deviation of goodput for five experiments, for a single connection using DART, SCPS-TP and selected TCP variants. It can be observed that BGAN results are consistent with IPSTAR results: DART and Hybla are the best choice for shared satellite links.

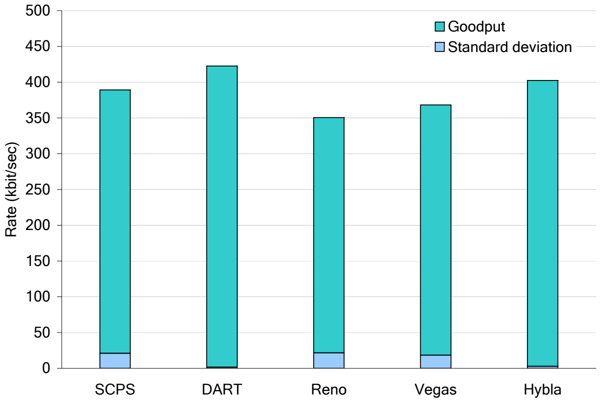

Figure 5. Average and standard deviation of goodput for five BGAN experiments, for a single connection using DART, SCPS-TP and TCP variants

To evaluate the responsiveness of selected protocols to congestion events, we perform experiments in which we introduce a UDP traffic spike of 30s duration mid way through the file download. Figure 6 presents a summary of results from IPSTAR experiments using the same non-compressible 10MB file.

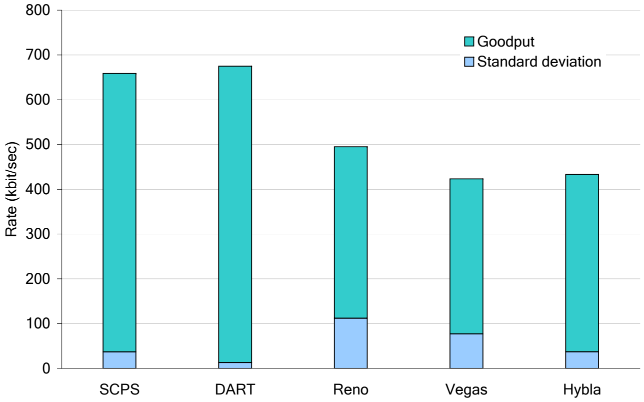

Figure 6. Impact of a UDP traffic spike on goodput; average goodput and standard deviation values from five IPSTAR experiments

It can be observed that DART both reacts to congestion and recovers from it well, while SCPS-TP, consistent with previous results, cannot cope with congestion.

A. *Evaluation of Additional Gains Achieved by Compression*

We perform additional experiments using files from the Canterbury corpus [14], which is commonly used to evaluate compression gains. It is important to note that compression is done within the transport mechanism, rather than in a store and forward way.

Table I shows the average goodput and the % increase compared to the average goodput of the non-compressible file of the same size, for ten IPSTAR and five BGAN experiments.

Combined E.Coli, bible.txt and kennedy.xls files were downloaded using the experimental setup described in Section IV.

The advantages of compression are clearly demonstrated by the results. For both BGAN and IPSTAR links, the available link data rate as perceived by the application is significantly increased compared to using DART congestion control and scheduling with no compression, i.e. 2.8 times for IPSTAR and 3.4 times for BGAN. The overall increase in goodput for the transport protocol using DART and with integrated compression, compared to standard TCP Reno, is 3.4 times for IPSTAR and over four times for BGAN.

As the amount of data actually transmitted over the link is reduced, for satellite services in which the charges are volume based (like BGAN) the increase in application rate is equivalent to the decrease in cost. For the case of the Canterbury files used in the experiments, the decrease in cost on BGAN compared to the case with no compression and the same congestion control mechanism is over three times.

Table I
COMPRESSION GAIN AND AVERAGE GOODPUT FOR THE CANTERBURRY CORPUS FILES

	IPSTAR	BGAN
Compression gain	286%	337%
Goodput (kbit/sec)	2395	1424.5

VI. FAIRNESS TO OTHER FLOWS

To evaluate fairness of multiple flows, we use Jain's fairness index [19]. We evaluate both self-fairness, i.e. fairness to data flows using the same transport protocol and fairness to standard TCP Reno. The fairness index is calculated for two flows sharing the IPSTAR link and presented in Table II.

Table II
FAIRNESS INDEX VALUES FOR TWO FLOWS

Fairness Index	DART	SCPS-TP	Vegas	Hybla	Reno
Self fairness	0.9995	0.92	0.998	0.996	0.91
TCP fairness	0.999997	0.997	0.63	0.66	0.91

It can be observed that our proposal is reasonably fair to both flows using the same congestion control and to TCP. Although it does not use the same congestion control mechanism, it is designed to respond to congestion events by lowering the rate, therefore producing a similar result.

VII. CONCLUSIONS AND FUTURE WORK

We have evaluated the performance of a new congestion control and scheduling mechanism against the commonly used TCP variants on two well established commercial satellite services. We have demonstrated the advantages of the proposal and benefits of the integrated data compression scheme, in terms of increased data rate and reduced data volume and corresponding cost. We have also shown that the proposal is reasonably fair to flows using the same mechanism and to TCP. Our plans for further improvements include enhancements of the compression engine to achieve higher compression gains.

VIII. ACKNOWLEDGEMENT

This research work has been supported by funding from National ICT Australia (NICTA). Authors would also like to thank IPSTAR Australia and Inmarsat for their continuing support.

REFERENCES

[1] C. Caini, R. Firrincieli, M. Marchese, T. de Cola, M. Luglio, C. Roseti, N. Celandroni, and F. Potort, "Transport layer protocols and architectures for satellite networks," in *International Journal of Satellite Communications and Networking*, Oct. 2006.

[2] C. Caini, P. Cornice, R. Firrincieli, and D. Lacamera, "A DTN approach to satellite communications," in *JSAC*, June 2008.

[3] C. Caini et al., "TCP Live Experiments on a Real GEO Satellite Testbed," in *ISCC 2007*, July 2007.

[4] CCSDS 714.0-B-2, "Consultative Committee for Space Communications, Recommendation for Space Data System Standards, Space Communications Protocol Sspecification Transport protocol (SCPS-TP)," Tech. Rep., Oct. 2006.

[5] Inmarsat BGAN, "Global voice and broadband data," Tech. Rep., Feb. 2009, http://www.inmarsat.com/.

[6] IPSTAR, http://ipstar.com/.

[7] WildBlue, http://wildblue.com/.

[8] Tolly Group, "Expand Networks Accelerator 4800 Series: WAN Compression Evaluation," Tech. Rep., Nov. 2004, No. 204115.

[9] SCPS-TP Reference Implementation, http://globalprotocols.com/.

[10] 7-ip Instant Communications, www.7-ip.com.

[11] SkipWare, http://globalprotocols.com/.

[12] Igor Pavlov, "LZMA Software Development Kit," www.7-zip.org/.

[13] GNU Wget, http://www.gnu.org/software/wget/.

[14] M. Powell, "Evaluating lossless compression methods," Tech. Rep., Feb. 2001, http://corpus.canterbury.ac.nz.

[15] G. Sarwar, R. Boreli, E. Lochin, and G. Jourjon, "Improvements in DCCP congestion control for satellite links," in *IWSSC*, Toulouse, France, Oct. 2008.

[16] W. Thesling et al., "Two Way Internet over IPSTAR Using Advanced Error Correctiona and Dynamic Links," in *AIAA Int. Comm. Satel. Syst. Conf.*, Montreal, Canada, May 2002.

[17] D. Bath, X. Bouthors, P. Febvre, K. Eckstein, E. Kristiansen, and J. Castro, "Extending Inmarsat BGAN to efficiently support the 3GPP Multimedia Broadcast / Multicast Service," in *AIAA Int. Comm. Satel. Syst. Conf.*, San Diego, USA, June 2008.

[18] ETSI EN 301 790 V1.4.1, "Digital Video Broadcasting (DVB); Interaction channel for satellite distribution systems," Tech. Rep., Sep. 2005.

[19] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*, Wiley-Interscience, New York, NY, Apr. 1991, ISBN:0471503361.