



HAL
open science

Improvements in DCCP congestion control for satellite links

Golam Sarwar, Roksana Boreli, Guillaume Jourjon, Emmanuel Lochin

► To cite this version:

Golam Sarwar, Roksana Boreli, Guillaume Jourjon, Emmanuel Lochin. Improvements in DCCP congestion control for satellite links. International Workshop on Satellite and Space Communications 2008 (IWSSC'08), Oct 2008, Toulouse, France. 5p. <hal-00361342>

HAL Id: hal-00361342

<https://hal.science/hal-00361342v1>

Submitted on 15 Feb 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Improvements in DCCP congestion control for satellite links

Golam Sarwar^{*,†,‡}, Roksana Boreli^{*,‡}, Guillaume Jourjon^{*}

^{*} National ICT Australia Ltd, Australia,

[†] University of NSW, Kensington, Australia,

[‡] 7-ip, Eveleigh, Sydney, Australia,

forename.surname@nicta.com.au

Emmanuel Lochin

Université de Toulouse,

DMIA-ISAE, LAAS-CNRS, France,

emmanuel.lochin@isae.fr

Abstract—We propose modifications in the TCP-Friendly Rate Control (TFRC) congestion control mechanism from the Datagram Congestion Control Protocol (DCCP) intended for use with real-time traffic, which are aimed at improving its performance for long delay (primarily satellite) links. Firstly, we propose an algorithm to optimise the number of feedback messages per round trip time (RTT) rather than use the currently standard of at least one per RTT, based on the observed link delay. We analyse the improvements achievable with proposed modification in different phases of congestion control and present results from simulations with modified ns-2 DCCP and live experiments using the modified DCCP Linux kernel implementation. We demonstrate that the changes results in improved slow start performance and a reduced data loss compared to standard DCCP, while the introduced overhead remains acceptable.

I. INTRODUCTION

Despite of an undeniable success of the TCP service as the primary means to carry reliable data over IP networks, an increasing number of new multimedia applications have found TCP too limiting. This is one of the reasons that pushed IETF to propose a new transport protocol called Datagram Congestion Control Protocol (DCCP) [1] which is particularly adapted to carry multimedia traffic that does not need a reliable service.

DCCP includes multiple congestion control algorithms which can be selected based on the user applications and which are identified by the Congestion Control ID (CCID).

We consider improvements for CCID3 [2], the TCP-Friendly Rate Control (TFRC) algorithm which is suitable for traffic with smooth changes in sending rates, such as telephony or video streaming. The CCID3 is based on the TCP throughput equation [3] and is designed to be reasonably fair when competing with TCP flows.

There has been an increasing deployment of satellite networks in recent times [4], [5], [6] and an increasing trend towards their use in residential and small business environments, rather than the previously used dedicated environments i.e. governmental, military or oil/gas/mining remote sites. This increases the applicability of new protocols like DCCP for use in those networks, as the networks will have an increasing amount of multimedia and real time traffic and the primary purpose of introducing DCCP was to better handle such traffic and provide a degree of transmission control which is not integrated in UDP. As the performance of TCP over satellite

and wireless links has been extensively evaluated with a number of proposed protocol enhancements, [7] it is likely that the DCCP algorithms (which reproduce a congestion control similar as the one of TCP) should also have performance difficulties over these networks.

In this paper, we show that the current feedback messages frequency is suboptimal in the context of long delay network and that a dynamic evaluation of this frequency can improve the goodput of the DCCP/CCID3 connection. Thus, we propose an algorithm to determine the optimum number of feedback messages (as opposed to using the currently standard of at least one per round trip time RTT) in DCCP/CCID3 (DCCP with TFRC congestion control). This algorithm is based on the observed link delay and is dynamically applied in the DCCP receiver. We first analyse the improvements achievable with our modifications in different phases of TFRC congestion control and present results from ns-2 simulator. In addition to simulation, we drive real measurements on a live satellite link by implementing our proposal inside the Linux kernel DCCP/CCID3 . We demonstrate that the modification results in a reduced data loss and higher data rates while the overhead introduced remains acceptable. In a previous work [8], we have introduced the idea of increasing the number of feedback messages and have presented results of experimental evaluation of the performance of DCCP/CCID3 for example values of the increased number of feedback messages. In this paper, we push further this idea with the proposition of a dynamic algorithm able to compute efficiently the optimal number of feedback messages.

The rest of the paper is structured as follows: section II provides an overview of the related work. Section III presents the proposed changes to the TFRC rate control. Section IV presents the simulation and experimental setup and Section V analyses how the changes affect the different stages of congestion control. Section VI presents results of live satellite tests and additional analysis. Conclusions and a discussion of future work is presented in Section VII.

II. RELATED WORK

DCCP/CCID3 [1], [2] and its variant for small packet sizes, DCCP/CCID4 [9], use TFRC congestion control. TFRC is an equation-based congestion control mechanism in which

the sender and receiver jointly compute an optimal sending rate based on the monitored network conditions. TFRC uses equation (1):

$$X = \frac{s}{(RTT \cdot \sqrt{\frac{p \cdot 2}{3}} + RTO \cdot \sqrt{\frac{p \cdot 27}{8}} \cdot p \cdot (1 + 32 \cdot p^2))} \quad (1)$$

Where:

- X is the transmit rate in bytes/second;
- s is the packet size in bytes;
- RTT the round trip time in seconds;
- p is the loss event rate, between 0 and 1.0, to represent the number of loss events as a fraction of the number of packets transmitted;
- RTO is the TCP retransmission timeout value in seconds.

TFRC results in a smoother sending rate and lower data throughput variations compared to TCP, which makes it suitable for multimedia streaming and VoIP applications. The reliance on receiver feedback, as shown in [10], makes TFRC respond slower to changes in available bandwidth compared to TCP in the case of long delay. Furthermore, the DCCP/CCID3 and CCID4 performances in the context of satellite delay have been recently studied [11]. In order to palliate some drawbacks of TCP and TFRC, the authors evaluate in [11], [12], using ns-2 [13] simulation, the impact of Quick Start [14] and Fast Restart [15] transport protocol mechanisms on the overall DCCP performance and show that these new mechanisms do not provide a consistent improvement to poor TFRC performances over a long delay network.

In this paper, we propose an alternative method to improve the performance of DCCP/CCID3 which uses a combination of already available parameters to dynamically adjust the way DCCP control behaves in long delay networks. As the equation based (1) mechanism uses measured RTT and loss and the measurement is done within each feedback period, the whole feedback-based algorithm is both slow and late to respond to network changes in long delay networks. Our proposal [8] to increase the number of feedback messages per RTT has shown significant improvements on live satellite links. However, our choice of experimental values was heuristic and to realise the potential of proposed changes we needed to characterise the optimum choice of parameter value and come up with an algorithm which can be applied dynamically to produce the optimum value.

III. PROPOSED IMPROVEMENTS

This section presents proposed improvements. In the following section, we analyse the effect of the proposed improvements in different stages of TFRC congestion control.

We propose to increase the number of feedback messages based on the measured delay and taking into consideration the received data rate and packet size. We argue that in a rate based algorithm, changes can be applied at any time, rather than needing to follow the logic of a window based algorithm where the changes are applied once every RTT. Care needs to be taken to prevent oscillations, as has been identified in [3].

We define an observation interval O as the length of time used for all calculations on which receiver feedback is based, with $O = RTT$ in the standard TFRC implementations and $O = RTT/N_{Fb}$ with N_{Fb} being defined as the number of feedback messages per RTT in our proposal. The optimum N_{Fb} is calculated as:

$$N_{Fb} = \min(\max(\text{round}(\frac{RTT}{RTT_{ref}}), 1), \frac{RTT * X_{rec}}{s}) \quad (2)$$

Where: X_{rec} is the received rate in bytes/second; RTT_{ref} is the reference (standard) link delay; s the mean size packet; and round : a function that rounds RTT/RTT_{ref} to the nearest integer.

We note that the receiver can only measure the received data rate and the average loss interval used for calculating the loss event rate p based on the fully received packets, and as a minimum needs to receive and process one packet in O . This results in a lower bound for the number of feedback messages per RTT in equation (2).

Please note that all the parameters are already calculated as part of the TFRC congestion control algorithm.

IV. SIMULATION AND EXPERIMENTAL SETUP

We use ns-2.32, patched to include the proposed modifications inside the official TFRC implementation. The simulation parameters include RTT, error rate and a congestion bottleneck equivalent to a representative satellite link bandwidth. We have chosen the parameters to closely match the IPSTAR satellite link we have at NICTA Laboratory in Sydney, Australia. Other than the proposed improvements, the standard DCCP/CCID3 default parameters, as recommended in [16], are used for all simulation and live tests. The DCCP packet size used in all experiments is 1000 bytes and the reference RTT_{ref} used is 100msec which is consistent with measurements in [17].

Previous experimental results [8] characterised the IPSTAR satellite network, which we consider a good representative of the growing number of IP based satellite services. IPSTAR uses shared access over radio channels and consequently can have both congestion and errors on the link. The network RTT and loss characterised during our long term experiments show an average RTT of around 1200msec and a packet error rate (PER) ranging from 0.24% to 0.42%. Other satellite networks e.g. DVB-RCS [18] have a specified BER of lower than 10^{-9} , resulting in a similar lower bound for PER for the packet size we have used in experiments and analysis.

The experimental setup is presented in Figure 1. For live satellite link tests, the DCCP-capable application using a DCCP protocol client is connected to the DCCP protocol server and the application server, through a satellite link which is connected to the public Internet through a local (Australian based) gateway. The ns-2 configuration includes an intermediate router and two links, one of which has a limited bandwidth and delay to match the representative link parameters.

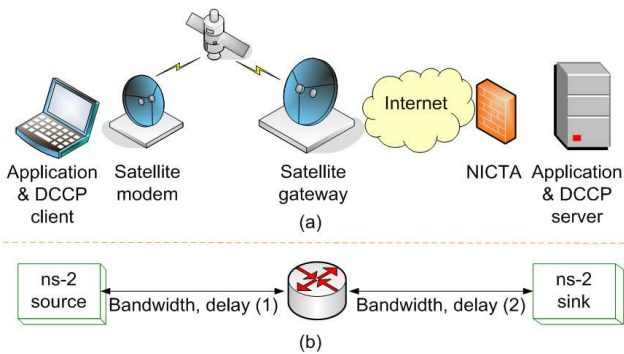


Figure 1. Experimental setup: (a) for live tests and (b) ns-2 simulation

V. ANALYSIS

This section presents analysis of the potential gains achievable with the modified TFRC algorithm. We demonstrate that the larger number of (accurate) feedback messages will result in a more responsive control mechanism and illustrate the analysis with results from ns-2 simulations.

A. Slow start

In the slow start phase, the sender increases the rate based on receiver feedback. For long delay links, the rate of growth can be very slow as the feedback is provided once every RTT, which is very likely to be less than once per second (as per observed RTT times on IPSTAR).

The minimum delay (D_{min}) with which the sender will receive feedback from the receiver can be viewed as:

$$D_{min} = RTT + O + t_{delay} \quad (3)$$

with t_{delay} being the time elapsed between the receipt of the last data packet and generation of this feedback report in the receiver as per [3]. For an observation interval of RTT, the sender will therefore make a rate adjustment based on the receiver feedback approximately every $2 * RTT$.

In our proposal, the observation interval is shortened to RTT/N_{Fb} . From the point in time where the receiver starts seeing a measurable amount of received data in the observation interval, the minimum delay with which the sender will receive feedback from the receiver will now be:

$$D_{min} = \frac{N_{Fb} + 1}{N_{Fb}} * RTT + t_{delay} \quad (4)$$

Applied in a feedback loop in which the value of the sender rate X is adjusted on receiving every feedback, the improvement in a time period t (assuming the starting rate X of one packet per second as per) can be approximated by:

$$X_t = s * 2^{\frac{t}{2 * RTT}} \quad (5)$$

with standard rate control and

$$X_t = s * 2^{\frac{t * N_{Fb}}{(N_{Fb} + 1) * RTT}} \quad (6)$$

with N_{Fb} feedbacks per RTT.

B. Congestion avoidance phase

Once the receiver detects the first loss event of the connection, TFRC mechanism enters into the congestion avoidance phase. During this phase, the sending rate is computed using the equation (1), with packet loss rate p and RTT as parameters. The p value is estimated at the receiver based on loss interval durations (number of packets between two loss events), using a weighted moving average algorithm which includes a history of previous loss interval durations and the number of packets received in the current error free interval. The value of RTT is computed at the sender using the receiver feedback on the recorded one-way packet delay and is also a weighted average over the duration of the observation interval.

In the error free periods, we can analyse the impact of the increased frequency of feedbacks on the resulting sender rate separately for the loss event rate estimation and for RTT estimation.

The packet-loss-rate estimation algorithm is the reason why increasing the number of feedback messages will have a positive effect on the estimation of the loss event rate in periods with no errors or congestion, as the increasing amount of data received, with more frequent observations, will be recorded faster and the loss event rate will therefore be reduced more quickly which will in turn result in a higher value of sender rate.

Having more frequent feedback with shorter observation intervals results in reported RTT values which are closer to the RTT values for individual packets. This is particularly important when RTT values increase, which is an arguable indication of getting closer to congestion. The variation of RTT will result in accordingly adjusted values of sender rate. In a dynamic environment, as will be shown in Section VI, the RTT changes will have a greater impact on the equation (1) based sender rate during the periods with no errors, as RTT in these periods varies more than p .

C. Demonstration of performance improvement in ns-2

This section presents the improvement achieved with increasing the number of feedback messages per RTT in ns-2.

Figure 2 shows result of ns-2 simulation of the DCCP/CCID3 performance with standard and with dynamic feedback, with RTT of 1sec on the nominal satellite link with a 1Mbit/s downlink rate, patched to include the same number of errors and RTT values on exiting slow start phase. In this experiment, it can be observed that there is a 14sec improvement with the dynamic feedback algorithm the standard algorithm when exiting the slow start. It can also be observed that the proposed modification results in about 20 sec improved loss recovery after the slow start phase. Please note that all the TFRC safeguards in regards to maximum rate increase as defined in [3] are still followed.

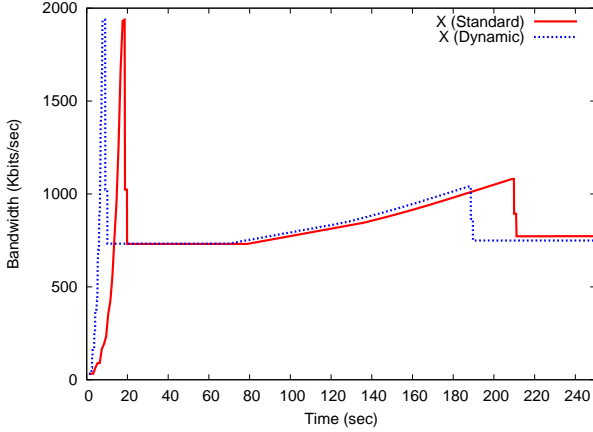


Figure 2. Slow start and error recovery on a representative satellite link, ns-2 simulation, standard and dynamic feedback DCCP/CCID3

The on-going congestion avoidance is more complex and the simplistic ns-2 simulation could not demonstrate the dynamic range of the RTT and error rate changes, therefore we have based the analysis on the results of experiments done on live satellite links, presented in the section below.

VI. RESULTS OF LIVE TESTS AND PERFORMANCE ANALYSIS

We have performed extensive experiments over the IPSTAR satellite network and the below figures show representative receiver rates for the standard TFRC and for the dynamic feedback algorithm.

The experiments were done a number at different times of the day (as the IPSTAR network peak congestion times coincide with business hours) and with different durations. A summary of the results from 50 download experiments are presented in Table I. The loss is the actual packet loss rate for each experiment (not the loss event rate used in TFRC rate control).

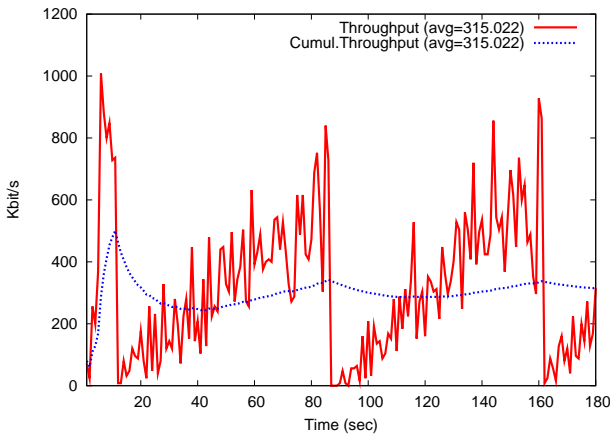


Figure 3. Measured throughput with the standard TFRC (1 feedback message per RTT by default)

It can be observed that the average download rates are on consistently higher by 33% and that the majority of the results

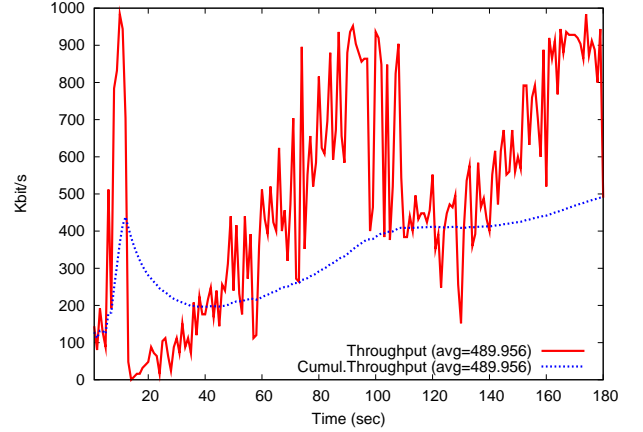


Figure 4. Measured throughput with the dynamic feedback algorithm

Table I
DCCP DOWNLOAD PERFORMANCE ON IPSTAR

	Avg. rate download (kbit/s)	Std. Dev. download (kbit/s)	Avg. Loss (%)
Standard TFRC	372	35.6	2.9%
Dynamic Fb TFRC	529	44.1	1.9%

are consistent, i.e. higher rates are achieved with the dynamic feedback version.

To aid with analysis, we also show the representative loss event rate p values calculated at the receiver and sender-observed RTT values in the Figures 5 and 6. As the slow start is exited when the receiver detects the first error, we can also use the p values to observe the difference in slow start phase for both algorithms.

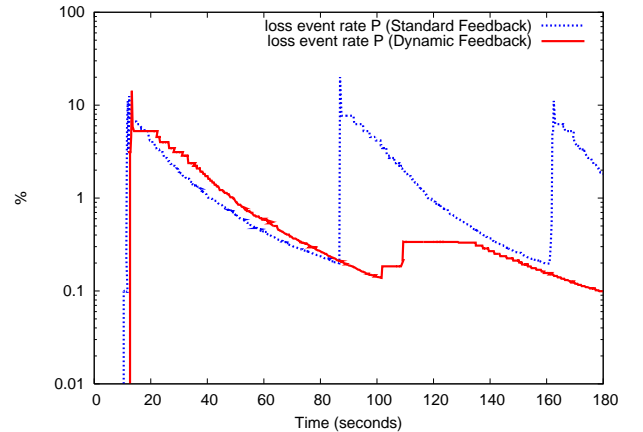


Figure 5. Loss event rate measured at the receiver for standard and dynamic feedback DCCP/CCID3

A. Slow start phase

It can be observed, consistent with the observations in the controlled environment of ns-2, that the slow start phase is also exited earlier for the dynamic feedback algorithm. This occurs close to 3 seconds earlier for dynamic feedback TFRC.

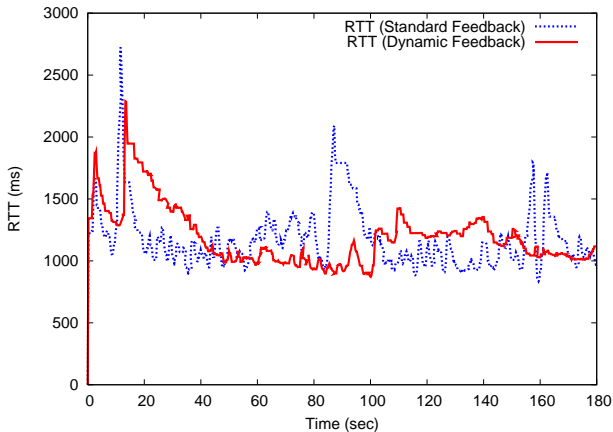


Figure 6. RTT measured at the receiver for standard and dynamic algorithms

B. Congestion avoidance phase

The on-going congestion control, as indicated by the results in Table I, shows a marked improvement for the dynamic feedback algorithm. The values of p and RTT can be directly correlated with received rates, which logically follows from the formula based algorithm. E.g. the error events recorded around 82 and 160sec from Figure 5 correspond to lowered rates recorded in the standard DCCP in Figure 3. For the dynamic feedback, higher RTT values around 90 sec from Figure 6 result in lower rate values in the same period in Figure 4, followed by a more noticeable rate drop from error events occurring between [100, 110] sec in Figure 5. We note that while the relation of sender rate to p is clearly reactive, i.e. after reported error events, the influence of RTT can be seen as both predictive and reactive i.e. the sender will reduce the rate based on higher values of RTT, which occurs both in error free intervals and after error events.

C. Discussion on the performance results

We believe the main benefit of the dynamic feedback mechanism in the congestion avoidance phase to be the increased accuracy of the RTT measurements, which are arguably an indicator of congestion, as demonstrated by the proposed TCP Vegas [19]. While a higher increase in the number of feedback messages may further improve the adjustments to dynamic network conditions and provide highest possible gains, it is important not to overload the return channel with control messages. The proposed dynamic feedback algorithm adjusts the number of feedback messages to the level of feedback traffic equivalent to what DCCP/CCID3 would generate on a standard internet link.

VII. CONCLUSIONS AND FUTURE WORK

We have proposed a change in the DCCP/CCID3 TFRC congestion control mechanism to improve its performance for long delay links. Our proposal is an algorithm to increase the number of feedback messages based on the link delay parameter. As the link delay is already measured within the DCCP protocol, the proposal does not significantly increase

the complexity of the control mechanism. We have analysed the performance of the proposed modifications and present results from ns-2 simulations and tests over a live satellite link. We have demonstrated that the proposed changes results in improved performance of DCCP/CCID3, while the increased overhead introduced remains acceptable.

We plan to further investigate DCCP over satellite and consider a TFRC-Vegas-like modification to rate control in order to predict and minimise error events.

VIII. ACKNOWLEDGEMENT

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

REFERENCES

- [1] E. Kohler, M. Handley, and S. Floyd, "Datagram Congestion Control Protocol (DCCP)," IETF, Request For Comments 4340, Mar. 2006.
- [2] S. Floyd, E. Kohler, and J. Padhye, "Profile for DCCP Congestion Control ID 3: TRFC Congestion Control," IETF, Request For Comments 4342, Mar. 2006.
- [3] M. Handley, S. Floyd, J. Padhye, and J. Widmer, "TCP-Friendly Rate Control (TFRC): Protocol Specification," IETF, Request For Comments 3448, Jan. 2003.
- [4] "INMARSAT BGAN," <http://bgan.inmarsat.com/>.
- [5] "IPSTAR," <http://ipstar.com/>.
- [6] "Wildblue," <http://wildblue.com/>.
- [7] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," in *Proc. of ACM SIGCOMM*, Aug. 1996.
- [8] G. Sarwar, E. Lochin, and R. Boreli, "Experimental performance of DCCP over live satellite and long range wireless links," in *7th International Symposium on Communications and Information Technologies (IEEE ISCI 2007)*, Sydney, Australia, Oct. 2007.
- [9] S. Floyd and E. Kohler, "Profile for DCCP Congestion Control ID 4: the Small-Packet Variant of TFRC Congestion Control," IETF, Internet Draft, 2006.
- [10] J. Widmer, "Equation-based congestion control," Diploma Thesis, University of Mannheim, Germany, Feb. 2000. [Online]. Available: <http://www.icsi.berkeley.edu/widmer/tfrc/thesis.ps.gz>
- [11] A. Sathiseelan and G. Fairhurst, "Performance of VoIP using DCCP over a DVB-RCS Satellite Network," in *IEEE International Conference on Communications (ICC)*, Glasgow, UK, Jun. 2007.
- [12] A. Sathiseelan and G. Fairhurst, "Use of Quickstart for Improving the Performance of TFRC-SP Over Satellite Networks," in *International Workshop on Satellite and Space Communications (IWSSC)*, Leganes, Spain, Sep. 2006.
- [13] "ns-2 network simulator," <http://www.isi.edu/nsnam/ns/>. [Online]. Available: <http://www.isi.edu/nsnam/ns/>
- [14] S. Floyd, M. Allman, and A. Jain, "Quick-Start for TCP and IP - draft-ietf-tsvwg-quickstart," IETF, Internet Draft, 2006.
- [15] S. Floyd and E. Kohler, "Faster Restart for TCP Friendly Rate Control (TFRC) - draft-ietf-dccp-tfrc-faster-restart," IETF, Internet Draft, 2006.
- [16] "Waikato University, DCCP site," <http://wand.net.nz/iam4/dccp/index.htm>.
- [17] L. J. Pietzuch, P. and M. Seltzer, "Supporting Network Coordinates on PlanetLab," in *Proc. of IWORLDS'05*, San Francisco, CA, Dec. 2005.
- [18] ETSI EN, "Digital Video Broadcasting (DVB); Interaction channel for satellite distribution systems," Tech. Rep., Sep. 2005, technical report 301 790 V1.4.1.
- [19] Lawrence S. Brakmo and Sean W. O'Malley and Larry L. Peterson, "TCP Vegas: new techniques for congestion detection and avoidance," *SIGCOMM Comput. Commun. Rev.*, vol. 24, no. 4, 1994.