On the quality of VoIP with DCCP for satellite communications

Golam Sarwar, Roksana Boreli, Emmanuel Lochin

To link to this article: DOI: 10.1002/sat.1010
URL: http://dx.doi.org/10.1002/sat.1010

To cite this version: Sarwar, Golam and Boreli, Roksana and Lochin, Emmanuel *On the quality of VoIP with DCCP for satellite communications.* (In Press: 2012) International Journal of Satellite Communications and Networking, vol. 30 (n° 4). pp. 163-180. ISSN 1542-0973

On the Quality of VoIP with DCCP for Satellite Communications

Golam Sarwar[1,2], Roksana Boreli[1,2], Emmanuel Lochin[3]

[1] National ICT Australia Ltd, Australia,
[2] University of NSW, Kensington, Australia,
[3] Université de Toulouse; ISAE ; LAAS-CNRS; Toulouse, France

Abstract

We present experimental results for the performance of selected voice codecs using DCCP with CCID4 congestion control over a satellite link. We evaluate the performance of both constant and variable data rate speech codecs for a number of simultaneous calls using the ITU E-model. We analyse the sources of packet losses and additionally analyse the effect of jitter which is one of the crucial parameters contributing to VoIP quality and has, to the best of our knowledge, not been considered previously in the published DCCP performance results. We propose modifications to the CCID4 algorithm and demonstrate how these improve the VoIP performance, without the need for additional link information other than what is already monitored by CCID4. We also demonstrate the fairness of the proposed modifications to other flows. Although the recently adopted changes to TFRC specification alleviate some of the performance issues for VoIP on satellite links, we argue that the characteristics of commercial satellite links necessitate consideration of further improvements. We identify the additional benefit of DCCP when used in VoIP admission control mechanisms and draw conclusions about the advantages and disadvantages of the proposed DCCP/CCID4 congestion control mechanism for use with VoIP applications.

1 Introduction

Voice over IP (VoIP) has become a well established technology with a large number of operators offering services and an ever growing number of end users. A large proportion of VoIP services use the public Internet, rather than a globally reserved bandwidth. This presents a problem both for the VoIP quality and the congestion of public Internet, as VoIP most commonly uses the UDP protocol [1] which has no congestion control and has no concept of fairness to other flows sharing the same network.

To bridge the gap between UDP and TCP, which is a reliable transport protocol and is not suitable for real time traffic, a new transport protocol for multimedia applications, Datagram Congestion Control Protocol (DCCP), has been proposed by IETF [2]. The main driver for having congestion control in an unreliable transport protocol was fairness to TCP traffic, which constitutes majority of the traffic on any Internet link. DCCP includes multiple congestion control algorithms identified by the Congestion Control Identifier (CCID), so that the application not needing reliable transport can select the appropriate congestion control method. CCID3 [3] and it's small packet variant CCID4 [6] relies on the TCP-Friendly Rate Control (TFRC) algorithm which is suitable for traffic with smooth changes in sending rates, such as telephony or video streaming. TFRC, originally specified in [7] and updated in [37], is based on the TCP throughput equation and is therefore shown to be reasonably fair when competing with TCP flows. CCID3 is more suitable for streaming applications while CCID4 has been designed for applications with small packets like VoIP.

In geographically large countries with sparse population outside of the main centres like Australia, US or Canada and also in countries with a quickly growing infrastructure like India, there has been a number of new satellite network deployments in recent times [8],[9], [35]. These networks have an increasing amount of multimedia and real time traffic and need to be considered in developing new protocols like DCCP.

In this paper, we present results of an experimental evaluation of the performance of selected voice codecs which use DCCP/CCID4 with TFRC congestion control over the IPSTAR satellite network [9] which is operational in Australia and a large number of countries in Asia. We additionally present the measured characteristics of a commercial mobile satellite service, Inmarsat Broadband Global Area Network (BGAN) [10], although, due to the high cost of this service, we have not done an evaluation of the VoIP performance over it. To the best of our knowledge, this is the first study presenting results from live satellite network performance measurements of VoIP using DCCP. We measure the receiver packet loss and delay and evaluate the VoIP quality under different conditions of network load using the ITU E-model [11]. We also evaluate fairness to competing TCP traffic sharing the same network. To mitigate the perceived packet loss resulting from DCCP/CCID4, we propose modifications to the CCID4 algorithm, which, compared to an alternative proposal Quick-Start [12], does not require any additional link rate information from the receiver. We demonstrate that the modifications result in a significantly improved VoIP quality compared to the original CCID4, while preserving the fairness advantage that CCID4 has over UDP. We note that we do not evaluate over our real topology the performance of recent TFRC specifications [37] which indeed includes modifications which should improve the performance of DCCP-CCID3 and CCID4 over satellite (or any) links. Indeed, to date and to the best of our knowledge, none of these modifications are available yet inside the GNU/Linux kernel. However, we compare with ns-2 simulations in Section 9 these proposed modifications with our proposal.

The rest of the paper is structured as follows. Section 2 presents an overview of the most common method for VoIP quality evaluation and summarises the common voice codec parameters. Section 3 provides an overview of the related work and a description of the TFRC congestion control mechanism. Section 4 presents the experimental setup for live satellite tests. The following section presents initial experimental results. Section 6 outlines our proposed modifications to the CCID4 protocol, followed by the evaluation of VoIP quality for all experimental scenarios in Section 7. Section 8 demonstrates that our proposal continues to be fair to competing TCP flows, which was one of the main goals of introducing DCCP. We present a comparison with other approaches in Section9 and a discussion of the advantages and disadvantages of using DCCP for VoIP in Section 10. Finally Section 11 concludes and outlines plans for future work.

2 Voice Codecs and Quality of VoIP

This section presents a summary of the quality evaluation methods for the performance of voice codecs on IP networks and an overview of the commonly used voice codecs which we will use in our experiments.

Voice codecs process the digitised analogue speech signal and produce a data stream which consists of voice frames generated at regular intervals. Depending on the codec type, the output stream will consist of constant or variable frame size(s) and will have a corresponding data rate. The use of voice activity detection and discontinuous transmission (DTX) also influences the output data rate. The following table 1 lists details of codecs commonly used in IP telephony, G.711 [13], G.729 [14] and Speex [16].

Table 1: Voice codec parameters

	voice frame size (bits)	voice frame size (msec)	data rate (kbits/s)
G.711	1440	10	64
G.729	160	10	8
Speex	variable	20	variable

One or more voice codec frames may be included within a IP packet payload, with the resulting data rate being increased by the appropriate IPv4 or IPv6 header and the transport protocol overhead. The transport protocol overhead will depend on the protocol used: the commonly used UDP protocol, the DCCP protocol evaluated in this paper or other protocols which may be used, e.g. RTP [17][5]. The resulting stream of IP packets is carried by the network and received by the VoIP application, which will decode the received voice codec frames and forward them to the listening device. The choice of the voice codec will impact the quality perceived by the parties in the conversation and additionally the network will introduce delay and may not correctly deliver all the packets from the VoIP stream, also impacting the quality.

Mean Opinion Score (MOS) is an ITU defined quality metrics for voice [18]. As MOS is a subjective measurement which cannot be easily applied to a variety of changing network conditions, ITU has also defined an objective evaluation methodology, the E-Model [11], which allows for evaluation of the voice quality based on measurements of network parameters. This model, originally designed for telephony, is also used for IP VoIP traffic [19]. ITU recommendation G.1020 also defines the VoIP gateway-specific reference points and performance parameters [20].

The E-Model's quality metrics is the $R-Score$ which is computed as follows:

$$R = R_0 - I_s - I_d - I_e + A \qquad (1)$$

Where:

R_0, the basic signal-to-noise ratio and I_s, the simultaneous impairment factor, represent non-network related impairments which are also independent of the voice codec used;

I_d, the delay impairment factor, represents delay and echo related impairments;

I_e is the equipment impairment factor which is related to the specific voice codec's quality and ability to handle losses;

A is the advantage factor which relates to potential compensations for listeners who by necessity need to accept a lower quality as there is no other means of communication available. A is commonly used to compensate for the nominal quality loss when satellite links are used in telephony or VoIP.

For a chosen voice codec, ITU defines the parameters to be used in the $R - Score$ calculations and the above formula will consequently consist of a constant value and a variable part, calculated based on the measured packet loss rate and end to end delay.

In a recent paper [21], authors claim that this model can be improved for VoIP traffic over best-effort networks because of the large delay variation that might occur to IP packets. Although this paper refines the E-model in this context, the results obtained are in the same order of magnitude and we believe that the current E-model prevents overly optimistic results. Furthermore, recent VoIP study [22] shows that this model is accurate enough to result in a good estimation of the subjective audio quality obtained.

The relation between $R - Score$ and MOS values is given by the equation below:

$$MOS = 1 + 0.035R + 7.10^{-6}R(R-60)(100-R) \qquad (2)$$

The quality levels defined by ITU [11] relate to satisfaction of users conducting the telephony (or VoIP) conversation and are reproduced in Table 2.

In the following section, we describe the congestion control mechanism used in the DCCP protocol.

Table 2: Provisional guide for the relation between $R - Score$ and user satisfaction

R value (lower limit)	MOS value (lower limit)	User satisfaction
90	4.34	Very satisfied
80	4.03	Satisfied
70	3.60	Some users dissatisfied
60	3.10	Many users dissatisfied
50	2.58	Nearly all users dissatisfied

3 TFRC and Congestion Control for VoIP

This section presents an overview of the TFRC congestion control mechanism and a summary of related work.

DCCP/CCID3 [3] and DCCP/CCID4 [6] use TFRC [7], [37] congestion control. In the TFRC congestion control mechanism, the appropriate sending rate is computed based on the monitored network conditions. Sender regulates the transmitted rate based on the received feedback messages which include the measured received rate, delay and an approximation of the packet loss rate. TFRC congestion control includes, similar to TCP, a slow-start phase and a congestion avoidance phase.

In slow-start, before the sender has received any receiver feedback, the sender's transmit rate X is set to one packet per second [3]. After the receiver feedback is available, the sender's initial rate is calculated as per equation (3):

$$X = \frac{min(4 \cdot s, max(2 \cdot s, 4380))}{RTT} \tag{3}$$

Where RTT is the estimated round trip time in seconds and s the packet mean size in bytes.

During the remainder of the slow-start phase, the sender rate is increased with every received feedback, as per equation (4):

$$X = min(2 \cdot X, 2 \cdot X_{recv}) \tag{4}$$

Where X_{recv} is the receiver reported rate in bytes/second.

When the receiver reports the first error, TFRC enters the congestion avoidance phase, which uses equation (5) approximating the transmitted rate to what would be an equivalent rate of TCP under the same network conditions.

$$X = s \cdot f(p, RTT) \tag{5}$$

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

Where: p is the loss event rate.

RTO is the TCP retransmission timeout value in seconds.

CCID4 [6] differs from CCID3 only in the congestion avoidance phase. To calculate the sending rate X, in place of the packet size s in equation 5, CCID4 uses a fixed packet size of 1460 bytes modified by a header correction factor, according to the following equation (6):

$$X = 1460 \cdot \frac{s}{s + oh} \cdot f(p, RTT) \tag{6}$$

Where oh is the size of protocol overhead in bytes.

This ensures that the formula based rate is fair to both TCP and DCCP traffic, by using a common TCP packet size in place of the size of smaller VoIP packets.

The updated TFRC specification [37] includes modifications related to the slow-start and periods when the application has no data to send, as would be the case for VoIP with DTX in silence periods.

In previous work, the performance of VoIP with DCCP/CCID4 protocol over satellite links has been studied in [23], [12] using simulation, and over generic links in [22] using emulation. The authors propose the use of Quick-Start [4] and Faster Restart [24] mechanisms and show that these methods provide only a partial improvement to the DCCP performance over a long delay network. In this paper, our intention was to analyse DCCP/CCID4 performance in a more dynamic environment than what has been considered in previous work and to provide additional insight into how DCCP handles real VoIP traffic.

In our previous work, we have proposed a dynamic computation of the number of sent DCCP/CCID3 feedback messages as a function of the end-to-end connection delay [25]. This modification greatly improves the rate computation of DCCP/CCID3 over long delay links by increasing the responsiveness of TFRC. The latter is achieved by a more accurate and timely estimation of network parameters. In this previous work, we aimed at achieving a data rate comparable to TCP when sending or receiving a high rate data stream using CCID3. In this paper, we push further the idea of dynamic adjustments, based on observed network conditions, by investigating the parameters which will affect the perceived quality of VoIP carried by CCID4 over satellite links.

In the following section we present details of our experimental setup used to evaluate the VoIP performance with DCCP/CCID4.

4 Experimental Setup

Our experimental setup at the NICTA Laboratory in Sydney, Australia is presented in Figure 1. For all the VoIP tests, we use the IPSTAR satellite service, with data being transmitted from the client side by the satellite modem, through the IPSTAR satellite gateway and the public Internet to our gateway (server side). We also have an Inmarsat BGAN satellite service which we have used only to measure it's characteristics and demonstrate the applicability of our tests across different satellite services. For DCCP/CCID4, we use the experimental version of Linux kernel implementation, which we have modified to include our proposed changes as described in Section 6.

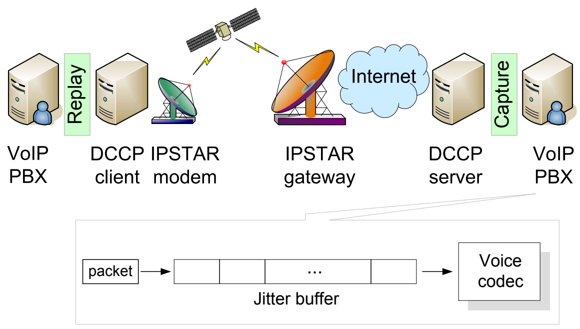

Figure 1: Experimental setup for live tests

The VoIP application used is Asterisk Private Branch Exchange (PBX) [26], with voice codecs commonly used in IP telephony. We use G.711 [13], Speex [16], with and without discontinuous transmission (DTX) and G.729 [14].

To have a fair comparison of quality with different codecs and different transport protocols, we use a pre-recorded sample of speech which is one side of a 10 min conversation. The analogue wave file is played into the VoIP PBX, encoded with the appropriate voice codec, transmitted using UDP and the Inter Asterisk Exchange (IAX2) protocol [26] and captured at the receiving end. All codecs send the encoded packets in 20msec intervals, i.e. G.711 and G.729 send two speech frames at a time. Our stream replicator application reads the UDP/IP payload and reproduces the timing and packet sizes of the VoIP packets. This data stream is transmitted using DCCP and captured at the receiving end for analysis. To produce examples of multiple conversations multiplexed into one data stream, we randomly start the pre-recorded conversation and we use the IAX2 multiplexing option.

Default DCCP/CCID4 parameters are used in all simulations, including the sender buffer size of five packets, consistent with other published work.

Previous experimental results [25] 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 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). Our subscription includes a 1Mbps downlink and 256kbps uplink data rate. 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} [27].

Inmarsat BGAN also uses shared bearers, with a nominal data rate of 492kbit/sec on both downlink and uplink [35]. Access control is implemented using TDM/TDMA [36]. 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. Other satellite networks of interest, e.g. DVB-RCS [28], 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 performed for the DCCP/CCID4 and the UDP protocol.

5 Initial Test Results and Performance Analysis

We perform a number of experiments over the IPSTAR satellite network, for different voice codecs and under different load conditions. All experiments are of 10min duration. Groups of experiments were performed at the same time to minimise the impact of IPSTAR congestion conditions on test results.

We measure the packet loss rate PLR, delay and jitter values at the DCCP receiver, which we will use to calculate the VoIP quality in Section 7. We also monitor parameters which contribute to the DCCP/CCID4 sender rate, including RTT, loss event rate p and receiver reported rate. A summary of the results from 10 IPSTAR experiments is presented in Table 3.

Table 3: Average packet loss rate values (%) for different codecs measured on IPSTAR link

Voice Codec & load	Data rate (kbit/s)	CCID4 (%) (%)	UDP (%)
G.711	80	2.01	0.15
G.729	22	1.24	0.1
Speex	average 25	1.84	0.1
Speex/DTX	variable	17.3	0.1
Speex,5 calls	average 96	2	0.15
G.711,12 calls	780	6.32	1.55

It can be observed that the packet loss with CCID4 has significantly higher values than the packet loss for UDP. This points to a potential for improvements, as the link (as demonstrated by the low UDP loss) can handle the VoIP traffic volume.

We also measure jitter, defined as the difference in the inter-packet gap of the successive VoIP packets. Jitter values observed in the experiments are shown in

Table 4, for the sender and the receiver side traffic. The sender side observations were included to highlight the regularity of the VoIP traffic, and confirm the expected timing of packets as per Table 1, although we note that the Asterisk PBX software occasionally introduces incorrect timing in the sender VoIP stream. The receiver side jitter is significantly higher, as the IPSTAR satellite link sends data in bursts [27]. Our observations show that the main source of jitter is the network rather than the DCCP congestion control algorithm, i.e. the average and maximum jitter values do not significantly differ between DCCP and UDP experiments. This will be further substantiated by additional experimental results presented in Section 7.

Table 4: Jitter values in milliseconds for all codecs

	Sender jitter (msec)		Receiver jitter (msec)	
	avg	max	avg	max
UDP	0.40	21.0	30.6	862.0
DCCP/CCID4	0.52	24.5	30.5	963.4

5.1 Performance Analysis

To assist with analysis of the error rate results, we consider the potential sources of packet loss at the input of the voice codec decoder. These include:

1. packet loss between the application and the transport protocol, P_{TP}, resulting from the inability of the transport protocol rate control to provide adequate sending rate to the application;

2. packet loss on the link, *PER*, which can be due to errors and/or congestion (related to the DCCP error event rate p);

3. the loss resulting from jitter, P_J, as the voice codec will consider all packets which arrive with incorrect timing as lost.

Jitter related losses at the VoIP application are counteracted by using a jitter buffer, which enables an even timing of speech frames at the voice decoder input. A jitter buffer of a specific length or duration will compensate to corresponding jitter values and consequently reduce jitter related errors. However, it will increase the delay and therefore reduce the VoIP quality (see Section 2). Choosing the jitter buffer length is therefore a trade-off which needs to consider other sources of packet loss and the link delay.

The *PLR* results summarised in Table 3 include both the link losses *PER* and the application-to-transport losses P_{TP} for DCCP. P_{TP} will only be applicable to DCCP, as UDP simply forwards application packets to the link.

Experimental data summarised in Table 3 indicates that only the experiments with G.711 had DCCP reported losses on the link and that all other losses were between the application and DCCP sender. Therefore, it can be concluded that the majority of losses are caused by the inability of DCCP/CCID4 protocol to provide a high enough sending rate to the VoIP application. Additionally, as there are no reported losses, CCID4 is operating in slow-start phase and never reaches the (expected and desired) congestion avoidance phase in which the equation (6) ensures TCP fairness.

To further verify our conclusion that the packet losses are primarily caused by the slow-start phase, we have also performed a series of experiments using the Linux Netem emulator [29] with different values of RTT. We use the IPSTAR equivalent data rate, as presented in Section 4, and there are no losses on the link. Figure 2 shows the resulting values of P_{TP} for different experimental scenarios. Please note that for UDP P_{TP} is always zero, as UDP will always transmit data presented by the application.

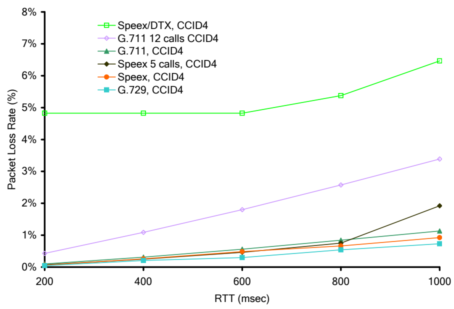

Figure 2: Packet loss rate for increasing RTT values, emulator, voice codecs: G.711, one and 12 calls; G.729; Speex, one and five calls

The following section presents our proposals to modify the DCCP/CCID4 protocol in a way which will enable better handling of the VoIP application traffic.

6 Improving DCCP/CCID4 for Long Delay Links

Experimental results indicate that significant VoIP packet losses occur in the slow-start phase, when there is an initial transition from silence to speech and, if DTX is used, after any silence periods. Therefore, we propose to modify the CCID4 rate control in the following way.

6.1 Novel Computation of the Sending Rate (CCID4-N)

In the first proposal, CCID4-N, we apply the existing CCID4 concept of replacing the measured packet size s by the equivalent packet size (of 1460 bytes modified by the header correction factor) to the slow-start phase. Consequently, in slow-start, the sender's starting rate will remain one packet per second, but with the packet size modified according to our proposal. After the receiver feedback is available, the initial rate will be calculated by the following formula, which will replace equation (3) in rate calculations:

$$X = \frac{4380}{RTT} \cdot \frac{s}{s + oh} \tag{7}$$

The proposed modification will increase the rate in initial stages, which will result in less packet loss in transitions between silence and speech.

6.2 Increasing the Number of Feedback Messages

We also apply the CCID3 modification proposed in [25], so that N feedback messages per RTT are used by the receiver in place of the default one feedback per RTT, when RTT is longer than a nominal value of e.g. 100msec. This increases the speed of rate growth in slow-start phase by applying formula (4) with increased frequency and, during the congestion control phase, provides more accurate values for changes of the RTT parameter used in formula (5) by more frequent measurements. The nominal RTT of 100msec represents a common value on the public Internet and has been adopted from the study of Internet traffic presented in [30].

The basic idea is 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 [37].

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:

$$N_{Fb} = \min\left(\max\left(round\left(\frac{RTT}{RTT_{ref}}\right), 1\right), \frac{RTT * X_{rec}}{s}\right) \tag{8}$$

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 (8).

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

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 and Inmarsat BGAN).

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

$$D_{min} = RTT + O + t_{delay} \tag{9}$$

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 [7]. 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:

$$D_{min} = \frac{N_{Fb} + 1}{N_{Fb}} * RTT + t_{delay} \tag{10}$$

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:

$$X_t = s * 2^{\frac{t}{2*RTT}} \tag{11}$$

with standard rate control and

$$X_t = s * 2^{\frac{t*N_{Fb}}{(N_{Fb}+1)*RTT}} \tag{12}$$

with N_{Fb} feedbacks per RTT.

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 (3), 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, the RTT changes will have a greater impact on the equation based sender rate (3) during the periods with no errors, as RTT in these periods varies more than p.

6.3 Second Proposal: CCID4-N_{100}

Our second proposal, CCID4-N_{100}, provides further optimisation for long delay links. We enhance the CCID4-N proposal by using a nominal value of RTT in place of the measured value in the slow-start phase. By using the RTT of 100msec in equation (7), the calculation of the rate in the slow-start phase becomes:

$$X = 43800 \cdot \frac{s}{s + oh} \tag{13}$$

The rationale of this idea is similar to TCP-Hybla [38] which suggests the necessary modifications to remove the performance dependence on RTT. Indeed, TCP-Hybla proposes to increase TCP reactivity over long-delay by taking a reference RTT denoted RTT_O and to modify the slow-start and the congestion avoidance phases as follows:

- slow-start : $cwnd = cwnd + 2k - 1$
- congestion avoidance : $cwnd = cwnd + 2k/cwnd$

where $k = RTT/RTT_O$. As for TCP-Hybla that sets an equivalent RTT to mimic a lower delay network, we propose the use of a nominal RTT value in place of the measured value. This approach is useful when such protocol is used e.g. over a PEP gateway and allows us to simplify the implementation of our proposal for a satellite PEP.

The sender rate during the slow-start phase is still increased based on the receiver reported rate in accordance with the equation (4).

The proposed modification will further increase the initial rate for long delay links, which should result in further reduction of packet loss in VoIP transmission. This will apply both to the start of any conversation and to DTX related silence period in which the voice codec may not transmit any frames. As we only insert a “fixed” low RTT in calculating the initial rate, if the VoIP rate is too high for the potentially congested link, the standard CCID4 mechanism will detect errors and take DCCP into congestion avoidance phase.

7 Performance Evaluation

VoIP quality will depend on the voice codec used, overall packet error rate which takes into account all sources of packet loss including the transport protocol, link and jitter and the total delay between the VoIP encoder and the decoded output of the VoIP codec decoder. In this section we first summarise the improvements in error rates achieved by the proposed modifications and present further experimental results for observed jitter values. We then evaluate the VoIP quality using the E-model described in Section 2, for a specific jitter buffer duration.

Table 5 presents the summary of the packet loss rate results from 10 IPSTAR experiments using the proposed CCID4 modifications, CCID4-N and CCID4-N_{100}. For comparison purposes we also include the CCID4 results from Table 3.

Table 5: Average packet loss rate values (%) for different codecs measured in experiments on IPSTAR link

	CCID4 (%)	CCID4-N (%)	CCID4-N_{100} (%)
G.711	2.01	0.44	0.15
G.729	1.24	0.08	0.1
SPEEX	1.84	0.15	0.1
SPEEX/DTX	17.3	0.16	0.1
SPEEX/5calls	2	0.34	0.15
G.711/12calls	6.32	3.76	1.5

It can be observed that *PLR* is significantly reduced by our proposals, with CCID4-N_{100} performing similarly to UDP. For higher VoIP data rates which are above the link rate, or in congested situations, our proposals will provide congestion control in the same way as CCID4.

Table 6 presents the summary of the jitter values observed in the experiments, for the sender and the receiver side traffic. The average jitter values shown here are similar to what is observed with UDP experiments (30.6msec) presented in Table 4. As there is no significant improvement in jitter reduction (this is consistent with the small DCCP buffer, although better VoIP rate handling is provided by our proposed improvements), our observations again show that the main source of jitter is the network.

Table 6: Jitter values in milliseconds for all codecs

	Sender jitter (msec)		Receiver jitter (msec)	
	avg	max	avg	max
CCID4-N	0.52	21.7	26.4	895.4
CCID4-N_{100}	0.45	21.0	28.2	798.0

7.1 Illustrating the Impact of the Dynamic Feedback Mechanism on the TFRC Performance

Before evaluating the whole performance of our proposal, we present some measurements that assess the benefit of increasing the number of feedback messages in the standard TFRC protocol (as defined by RFC 3448) with ns-2 and with the GNU/Linux implementation over IPSTAR.

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 [39]. 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.

Fig. 3 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 [7] are still followed.

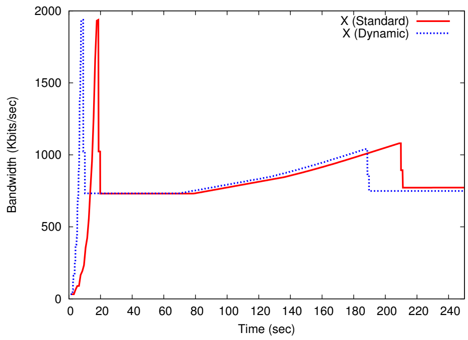

Figure 3: Slow start and error recovery on a representative satellite link, ns-2 simulation, standard and dynamic feedback DCCP/CCID3

The on-going congestion avoidance is quite complex and the simplistic ns-2 simulation could not demonstrate the dynamic range of the RTT and error rate changes, therefore we have performed extensive experiments over the IPSTAR satellite network. These experiments were done at different times of the day (as the IPSTAR network peak congestion times coincide with business hours)

Table 7: 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%

and with different durations. A summary of the results from 50 download experiments is presented in Table 7, for the standard TFRC and for the dynamic feedback algorithm. Note the loss shown in the table is the actual packet loss rate measured for each experiment (not the loss event rate used in TFRC rate control).

7.2 VoIP Quality

We now evaluate the VoIP quality for the voice codecs and network load scenarios from our experiments. The values of the $R - Score$ factor are calculated using the overall packet loss rate and the total delay, while MOS is then calculated from equation 2. To calculate the packet loss rate resulting from jitter, we choose a buffer size of 400 msec, as a compromise between adding delay and an increased packet loss. The resulting P_J of 0.01% used in calculations is the probability that a packet will have a jitter grater than the buffer value, averaged over all experiments. The jitter buffer size can be varied to further compensate for high jitter values, however that will not have an impact on the difference between the performance of DCCP and UDP as the measured jitter values for those protocols are very similar.

We note that the IPSTAR network has both a high jitter (requiring a significant jitter buffer size) and a high delay. To compensate for the delay impairment in the E-model calculations, we use an advantage factor of 40. This is higher than the recommended figure of 20, which is commonly used for carrier grade satellite telephony [11]. However, the maximum acceptable delay for toll quality satellite links is 400msec [31] [15], while we have almost double that value on IPSTAR and we believe that this justifies the use of this higher value. Additionally, we are interested in a relative comparison of VoIP quality between the DCCP and UDP protocols, when there is sufficient bandwidth to support the VoIP application rate and the advantage factor will not impact the relative comparison.

The $R - Score$ factor values are calculated with specific ITU parameter values, however, Speex is not an ITU codec and does not have the defined parameter values necessary for calculating the $R - Score$ factor. For the purpose of evaluating Speex quality, we roughly approximate the quality and error resilience of Speex codec to the corresponding parameters of the G.729 codec. We consider this approximation sufficiently accurate for the purpose of this evaluation, as the reported MOS score for the Speex codec used in our experiments is 4.1 [32], comparable to the MOS value of 4.18 for the G.729 codec, resulting from the E-Model calculations for the same network conditions.

Figure 4 shows the $R - Score$ factor for selected experimental scenarios on the IPSTAR satellite network for all the transport protocols considered.

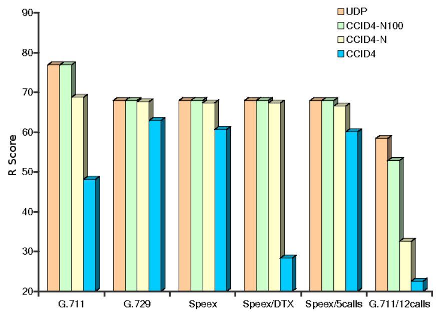

Figure 4: $R - Score$ factor for IPSTAR experiments, G.711, one and 12 calls, G.729, Speex, 1 and 5 calls, UDP, CCID4 and CCID4-N

To provide a clear view of the difference between the voice quality with UDP and with CCID4 variants, Figure 5 presents the degradation in MOS values compared to UDP.

It can be observed that the $R - Score$ factor values on IPSTAR network range between an acceptable 79 (with a corresponding MOS value of 3.9) for a G.711 call with either UDP or our proposal CCID4-N_{100}, to unacceptable values of below 50 for the same codec with CCID4 and even lower for Speex with DTX. Our proposals improve the voice quality compared to CCID4 for all cases considered and CCID4-N_{100} delivers voice quality similar to UDP for all but the highest number of calls.

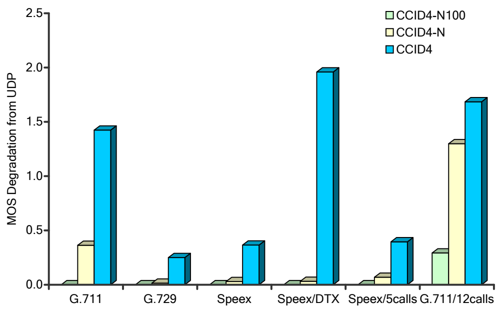

Figure 5: Degradation of MOS values from UDP for different congestion control mechanisms on the IPSTAR network

In the following section, we will evaluate the fairness of the proposed DCCP/CCID4 modifications.

8 Fairness to Other Flows

To evaluate fairness of multiple flows, we use Jain's fairness index [33].

Table 8: Fairness index values for two flows, TCP and CCID4 versions

Fairness Index	CCID4	CCID4-N	CCID4-N_{100}	UDP
G.711/12calls	0.9997	0.99997	0.99997	0.74
1Mbit/s data rate	0.985	0.98	0.98	0.5

As VoIP traffic is limited by the application, fairness can only be considered for VoIP streams which result from a number of parallel (multiplexed) calls which would require unfair capacity when sharing the link with other flows. We compare fairness to TCP of a VoIP data stream resulting from 12 parallel G.711 calls and additionally use a flow with rate equivalent to the nominal rate on the IPSTAR link. To illustrate the advantage of using DCCP, we also present the fairness results for UDP. The results of the fairness tests are presented in Table 8.

It can be observed that both our proposals and CCID4 are equally fair to a TCP flow and that UDP, as expected, takes all the bandwidth it requires regardless of other flows.

9 Related work and comparison with other approaches

To have a complete view of the topic, we also consider the recent specification proposed in RFC 5348 [37], which includes modifications that should also improve the performance of DCCP-CCID3 and CCID4 over a satellite link. The key area where RFC 5348 proposes improvements is the slow-start after idle periods. After such periods, the TFRC estimated sending rate is decreased as no data is sent. Thus, the algorithm that computes the current sending rate results in a corresponding decreased rate. Furthermore, when the idle period is too long, a slow-start is triggered.

TFRC has been designed to be TCP-friendly. As a matter of fact, changing the protocol behaviour after an idle period (for instance, by not using the standard slow-start) might introduce unfairness to TCP and modifying the algorithm for computing the rate during the idle period might impact the TCP compatibility. In a recent IETF draft [40], the authors e.g. propose to not reduce the sending rate during idle periods. However this approach is questionable as the `nofeedback` timer could expire because of an idle period, or because of data or feedback packets dropped in the network due to congestion events. Assessing the reason for timer expiry is difficult, with an equivalent complexity to that of differentiating losses from congestion on an erroneous link. Finally, we can argue that this approach distances the behavior of TFRC from TCP and we could get to the point where the proposed modifications, in the extreme case, result in a TFRC variant that behaves closer to the UDP protocol with a slow-start than to TCP.

A recent study [42] analysed the benefit brought by RFC 5348, as compared to RFC 3448. Figure 6 (by courtesy of the authors of [42]) presents the behaviour of RFC 3448 and RFC 5348 after an idle period, for a G.711 VoIP codec. The results presented in this figure allow us to understand the dynamics of these schemes. The authors have simulated a 64Kbps VoIP flow (with 160 byte packets) over a path with 250 ms one way delay, using ns-2. The application models a G.711 VoIP codec and the capacity is set to 2 Mbps, hence there is no congestion. The application becomes idle for a period starting at $t = 10$ seconds, with a duration of 10 seconds. This duration has been chosen to illustrate the effect of the sending rate reduction and restarting from the recover rate.

Table 9: Difference between RFC 3448 and RFC 5348

	RFC 3448	RFC 5348
Initial slow-start rate	1 packet/RTT	4 packets/RTT
Idle period recover rate	2 packets/RTT	4 packets/RTT
After idle period behaviour	Double sending rate	Double sending rate
Data-limited period behaviour	Limited by receiver rate $X = max(min(X_{calc}, 2 * X_{recv}), s/t_{mbi})$	Not limited by receiver rate but by an average of the two last values of recent X_{recv} values contained in X_{recv_set} $X = max(min(X_{calc}, recv_{limit}), s/t_{mbi})$ with $recv_{limit} = 2 * max(X_{recv\_set})$ depending on whether the feedback packet reports a new loss event or an increase in the loss event rate

As shown in Table 9, the biggest impact of the changes proposed by RFC 5348 is on the initial slow-start and the recovery rate value. The curves presented Figure 6 (by courtesy of the authors of [42]) clearly confirm that the major impact is given by the 4 packets/RTT rate at the initial slow-start and at the idle period recovery. These values obviously mitigate the impact of idle periods although the sender's response to a `nofeedback` timer has been changed (see data-limited period behaviour from Table 9) by preventing the sender to directly use the receiver's rate value. We can thus consider that the increase of both initial slow-start and idle recover rates results in the biggest performance impact. In their paper, the authors also investigate the Faster Restart (FR) mechanism [24] which quadruples the sending rate after idle periods (instead of doubling as in both RFCs) and increases the idle period recovery rate to 8 packets/RTT for small packet size (this value remains at 4 packets for standard TFRC packet size). Even with this increase, the authors highlight in their study and conclude that the addition of FR does not bring substantial performance improvement to RFC 5348 [42]. Indeed, the simulation scenario in Figure 6 clearly shows that for long idle periods (the ones that have the biggest impact on the performance of multimedia traffic), the main impact is obtained from the recovery algorithm which progress faster than the initial RFC3448. Note that between $t = [0; 5]$ seconds: RFC3448 with FR, RFC5348 and RFC5348 with FR behave similarly. We note that the initial window value is a current discussion item at the IETF and the latest discussions agree to set this value to ten packets [43]. Following the increased rate of the access links, it seems logical to increase the initial slow-start value which appears outdated. Thus, in the following year we may observe a complete change of the TFRC performance, if this value is confirmed and also adopted by TFRC.

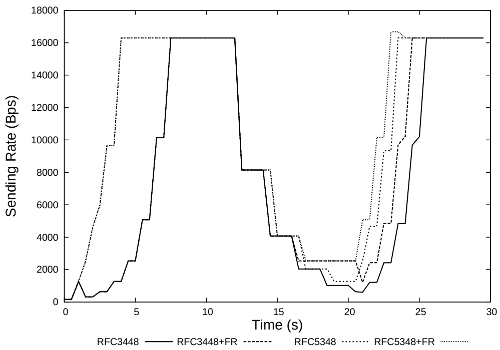

Figure 6: Sending rate dynamics of RFC 5348 and RFC 3448 with and without FR (by courtesy of the authors of [42])

Compared to all these works, our proposal and the context of use of this protocol differs in several points. First, we seek to use TFRC for VoIP traffic over satellite links. Although we use our scheme end-to-end, today, we do not observe a large deployment of the TFRC protocol. Indeed, DCCP [2] is not deployed in the most commonly-used operating systems. We do not think that this will change in the future and the current trend seems to encapsulate user-space

DCCP implementation inside UDP [41]. However, protocols that use shared satellite links also need to fairly share the available capacity as a function of the number of flows. Thus, using a congestion-controlled and unreliable protocol such as TFRC inside a PEP satellite gateway can be a potential solution. As a result, our proposal can be seen as for Faster Restart, as a complement to other schemes and improvements that have been done in a more generic context for the Internet. Compare to these proposals, we propose a novel calculation scheme for TFRC to fit long delay link requirements combined with: 1) an increase of the feedback frequency; 2) a novel computation applied to the recovery and initial slow-start periods that takes into consideration the link delay and not the number of packet sent per RTT (these proposed modifications and in particularly the feedback computation scheme, could also be added to RFC 5348 similarly than Faster Restart has been tested conjointly with both RFCs in [42]).

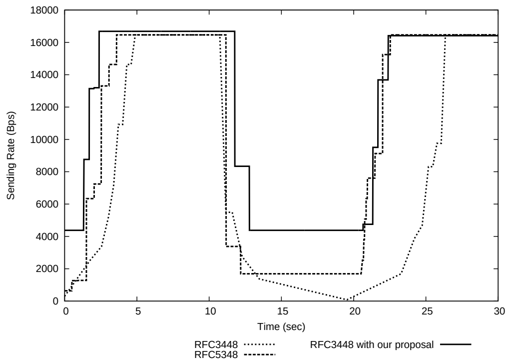

Figure 7: Sending rate dynamics of RFC 3448 and our proposal

Finally, Figure 7 presents the outcome of a similar experimental scenario as what was used to produce results shown in Figure 6. We reproduce the experiment using the ns-2 code for RFC 5348 of the authors of [42]. It can be clearly observed that our proposal does not contradict with the outcome of RFC 5348. As shown in Figure 6 and 7, our proposal helps the sending rate to grow faster at the beginning of the slow-start and multiple feedback messages help the rate growth after a data-limited period (comparable to silent period in VoIP) in slow-start.

10 The Pros and Cons of Rate Control

The main conclusion from the tests done with the original CCID4 is that in the vast majority of the cases, the low VoIP data rate does not create any congestion events, and as a consequence CCID4 continually operates in slow-start phase. In this phase, depending on the rate required by the VoIP application, in most cases

CCID4 cannot support the required rate and this results in significant packet loss, particularly on long delay links. Our proposed modifications significantly reduce the packet loss between the application and DCCP sender in a long delay environment. They also provide similar benefits in networks which have a lower delay, in enabling a faster slow-start and a higher minimum rate. However, DCCP still continues operating in slow-start and we need to look at scenarios where this will cease to be the case.

If we consider the benefits of rate control, it is beneficial to compare the differences in the environment in which the DCCP-CCID3 and CCID4 have been designed to operate. CCID3 is applicable to generic traffic which would normally be transmitted using UDP. The generic multimedia traffic may not have as stringent real time requirements as VoIP traffic does, therefore the rate changes dictated by DCCP may be more easily accommodated with applications using CCID3. VoIP traffic has an on-off pattern and in most cases requires a constant, or close to constant, data rate when the traffic is present on the link. So while congestion control is necessary for multimedia and VoIP traffic to regulate the global congestion on the Internet and enable fair sharing of bandwidth by different applications, it is also important to continue to support a good quality of those applications and to apply congestion control in a way which will provide benefits without unduly sacrificing the applications quality.

If we for the moment disregard the issues related to the slow-start phase, we can observe that the benefits of DCCP and rate control for VoIP applications are most apparent in cases when a number of simultaneous calls is being transmitted. The most beneficial use of DCCP would be in conjunction with call admission control (CAC), to regulate the number of simultaneous calls on the link in a multi-call scenario. Currently CAC is done by a combination of probing and known bandwidth limitations [34]. UDP has no ability to detects losses on the link and therefore cannot be used to aid CAC. As the number of calls increases and reaches the level where packets are lost, the DCCP measured loss rate can be used to trigger blocking of new calls by the VoIP application. To be able to use this feature, there needs to be a link between the allowable application rate (number of calls multiplied by the data rate for individual call) and the estimated fair rate on the link. This can be accomplished by a transport to application cross layer approach which we believe is needed to fully utilise the benefits of congestion control and DCCP-CCID4.

11 Conclusions and Future Work

We have evaluated the performance of DCCP/CCID4 on a live satellite link for a number of scenarios which include different voice codecs and a varying number of simultaneous VoIP calls. The main issue identified with using CCID4 for VoIP was in periods of transition from silence to speech, where in most cases CCID4 cannot support the required application rate and produces significant packet losses, particularly on long delay links. We have proposed modifications to CCID4 which mitigate this problem for the most common voice codecs, by enabling a faster slow-start, higher minimum rate and a more accurate parameter measurement resulting in a more responsive rate adjustment to the varying network conditions. Both our proposals require minimal changes to CCID4 specification and no interaction with other network components. They result in no loss of fairness to TCP traffic compared to the original CCID4. We note that the recently adopted changes to the TFRC specification will improve the results presented in this paper, however we believe that presenting experimental results over a live satellite link and the proposed simple modifications which could further close the gap in VoIP performance between terrestrial and satellite links are still valuable contributions.

With the proposed improvements and the inherent fairness it was designed for, we believe that CCID4 has a significant advantage over UDP. DCCP awareness of transport layer losses can also be used for VoIP call admission control to maximise the benefits of a rate based transport protocol and provide best possible call quality. In continuation of this work, we plan to further investigate DCCP aided call admission control and variable rate speech codec rate management for VoIP.

12 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, anonymous reviewers and Arjuna Sathiaseelan for several discussions on RFC 5348.

References

[1] J. Patel, "User Datagram Protocol" IETF, RFC 768, Aug. 1980

[2] E. Kohler, M. Handley, S. Floyd, "Datagram Congestion Control Protocol (DCCP)", IETF, RFC 4340, Mar. 2006

[3] S. Floyd, E. Kohler, J. Padhye, "Profile for DCCP Congestion Control ID 3: TFRC", IETF, RFC 4342, Mar. 2006

[4] G. Fairhurst, A. Sathiaseelan, "Quick-Start for the Datagram Congestion Control Protocol (DCCP)", IETF, RFC 5634, Mar. 2006

[5] C. Perkins, "RTP and the Datagram Congestion Control Protocol (DCCP)", IETF, RFC 5762, Apr. 2010

[6] S. Floyd and E. Kohler, "Profile for DCCP Congestion Control ID 4: TCP-Friendly Rate Control for Small Packets", IETF, RFC 5622, Aug. 2009

[7] M. Handley, S. Floyd, J. Pahdye, and J. Widmer, "TCP-Friendly Rate Control (TFRC): Protocol Specification", IETF, RFC 3448, Jan. 2003

[8] WildBlue, `http://wildblue.com/`

[9] IPSTAR, `http://ipstar.com/`

[10] Inmarsat, `http://www.inmarsat.com/`

[11] ITU T-REC-G.107, "The E-Model, a computational model for use in transmission planning"

[12] G. Fairhurst and A. Sathiaseelan, "Use of Quickstart for Improving the Performance of TFRC-SP Over Satellite Networks", in IWSSC, Leganes, Spain, Sep. 2006

[13] ITU T-REC-G.711, "Pulse code modulation of voice frequencies".

[14] ITU T-REC-G.729, "Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP)"

[15] ITU T-REC-G.114 Series G: Transmission systems and media, digital systems and networks, International telephone connections and circuits, General Recommendations on the transmission quality for an entire international telephone connection, one-way transmission time, May 2003

[16] J.M. Valin, "Speex: A Free Codec For Free Speech", in Linux Australia conference, Dunedin, New Zealand, Jan. 2006

[17] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", IETF, RFC 3550, Jul. 2003

[18] ITU T-REC-P.800, "Methods for subjective determination of transmission quality".

[19] R. G. Cole and J. H. Rosenbluth, "Voice over IP Performance Monitoring", ACM Computer Communications Review (CCR), vol. 31, no. 2, pp. 9-24, 2001

[20] ITU T-REC-G.1020, "Performance parameter definitions for quality of speech and other voiceband applications utilizing IP networks"

[21] A. Meddahi and H. Afifi, "Packet-E-Model: e-model for VoIP quality evaluation", Elsevier Computer Networks, vol. 50, no. 15, pp. 2659-2675, Oct. 2006

[22] H. Balan, L. Eggert, S. Niccolini, and M. Brunner, "An Experimental Evaluation of Voice Quality over the Datagram Congestion Control Protocol", in INFOCOM, Anchorage, USA, May 2007

[23] A. Sathiaseelan and G. Fairhurst, "Performance of VoIP using DCCP over a DVB-RCS Satellite Network", in ICC, Glasgow, UK, Jun. 2007

[24] S. Floyd and E. Kohler, "Faster Restart for TCP Friendly Rate Control (TFRC)", IETF, Draft #06, 2009

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

[26] Asterisk PBX, `http://www.asterisk.org/`

[27] W. Thesling, M. Vanderaar, M. Thompson, P. Hamilton, P. Panuwattanawong, and R. Gedney, "Two Way Internet over IPSTAR Using Advanced Error Correctiona and Dynamic Links", in AIAA Int. Comm. Satel. Syst. Conf., Montreal, Canada, May 2002

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

[29] Linux Netem emulator, `http://www.linuxfoundation.org/en/Net:Netem`

[30] P. Pietzuch, J. Ledlie, and M. Seltzer, "Supporting network coordinates on planetlab", in WORLDS'05: Proceedings of the 2nd conference on Real, Large Distributed Systems Berkeley, CA, USA: USENIX Association, 2005

[31] Telecom Regulatory Authority of India, "Regulation on Quality of Service for VoIP Based International Long Distance Service (First amendment)", File No.402-30/2001-FN (pt), Tech. Rep., Jan. 2004

[32] Speex codec, `http://www.speex.org/`

[33] 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

[34] Cisco Systems, Inc., VoIP Call Admission Control, Solutions Document, ver. 1.2, Tech. Rep., Aug. 2001

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

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

[37] M. Handley and S. Floyd and J. Pahdye and J. Widmer, "TCP-Friendly Rate Control (TFRC): Protocol Specification", IETF, RFC 5348, Apr. 2008

[38] G. Giambene and C. Caini and Nico C. Liberato and R. Firrincieli, "TCP Hybla Performance in GEO Satellite Networks: Simulations and Testbed", International Workshop on Satellite and Space Communications (IWSSC), Leganes-Spain, Sep. 2006

[39] Lawrence S. Brakmo and Sean W. O'Malley and Larry L. Peterson, "TCP Vegas: new techniques for congestion detection and avoidance", ACM Computer Communications Review (CCR), vol. 24, no. 4, 1994

[40] G. Fairhurst, I. Biswas, "Updating TCP to support Variable-Rate Traffic", IETF work in progress, no. 1, `draft-fairhurst-tcpm-newcwv`, Mar. 2011

[41] T. Phelan, G. Fairhurst, C. Perkins, "Datagram Congestion Control Protocol (DCCP) Encapsulation for NAT Traversal (DCCP-UDP)", IETF work in progress, no. 9, `draft-ietf-dccp-udpencap`, Jul. 2011

[42] A. Sathiaseelan, G. Fairhurst, "TCP-Friendly Rate Control (TFRC) for bursty media flows", Computer Communications, Volume 34, Issue 15, , Pages 1836-1847, 15 September 2011.

[43] N. Dukkipati, T. Refice, Y. Cheng, J. Chu, Jerry, T. Herbert, A. Agarwal, A. Jain, N. Sutin, "An argument for increasing TCP's initial congestion window", ACM Computer Communications Review (CCR), vol. 40, no. 3, pp. 26-33, 2010.