# Enhanced transport protocols for real time and streaming applications on wireless links

# ENHANCED TRANSPORT PROTOCOLS FOR REAL TIME AND STREAMING APPLICATIONS ON WIRELESS LINKS

Golam Sarwar (Babil)

A dissertation submitted in fulfilment of the requirements for the degree of
**Doctor of Philosophy (Ph.D.)**

**School of Electrical Engineering and Telecommunications (EE&T)**
**Engineering Faculty**
**University of New South Wales (UNSW), Australia**

in a cotutelle agreement with

**University of Toulouse, France**

August, 2013

# ORIGINALITY STATEMENT

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

Signed Golam Sarwar (Babil)
Date January 8, 2015

# COPYRIGHT STATEMENT

I hereby grant the University of New South Wales or its agents the right to archive and to make available my thesis or dissertation in whole or part in the University libraries in all forms of media, now or here after known, subject to the provisions of the Copyright Act 1968. I retain all proprietary rights, such as patent rights. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation. I also authorise University Microfilms to use the 350 word abstract of my thesis in Dissertation Abstract International (this is applicable to doctoral theses only). I have either used no substantial portions of copyright material in my thesis or I have obtained permission to use copyright material; where permission has not been granted I have applied/will apply for a partial restriction of the digital copy of my thesis or dissertation.

| | |
|---|---|
| Signed | Golam Sarwar (Babil) |
| Date | January 8, 2015 |

## AUTHENTICITY STATEMENT

I certify that the Library deposit digital copy is a direct equivalent of the final officially approved version of my thesis. No emendation of content has occurred and if there are any minor variations in formatting, they are the result of the conversion to digital format.

| | |
|---|---|
| Signed | Golam Sarwar (Babil) |
| Date | January 8, 2015 |

*Ohana* means family.
Family means nobody gets left behind, or forgotten.

— Lilo & Stitch

In the loving memory of my dearest grandmother Jahanara Begum who filled my childhood with unmatched love and affection. You and your magical house will always dearly be missed.

# ABSTRACT

Real time communications have, in the last decade, become a highly relevant component of Internet applications and services, with both interactive (voice and video) communications and streamed content being used by people in developed and developing countries alike. Due to the proliferation of mobile devices, wireless media is becoming the means of transmitting a large part of this increasingly important real time communications traffic. Wireless has also become an important technology in developing countries (and remote areas of developed countries), with satellite communications being increasingly deployed for traffic backhaul and ubiquitous connection to the Internet. A number of issues need to be addressed in order to have an acceptable service quality for real time communications in wireless environments. In addition to this, the availability of multiple wireless interfaces on mobile devices presents an opportunity to improve and further exacerbates the issues already present on single wireless links. Mitigation of link errors originating from the wireless media, addressing packet reordering, jitter, minimising the link and buffering (required to deal with reordering or jitter) delays, etc. all contribute to lowering user's quality of experience and perception of network quality and usability. Therefore in this thesis, we consider improvements to transport protocols for real time communications and streaming services to address these problems and we provide the following contributions.

To deal with wireless link issues of errors and delay, we propose two enhancements. First, an improvement technique for Datagram Congestion Control Protocol (DCCP) congestion control algorithm ID4 (CCID4) for long delay wireless (e.g. satellite) links, demonstrating significant performance improvements for Voice over IP (VoIP) applications. To deal with link errors, we have proposed, implemented and evaluated an erasure coding based packet error correction approach for Concurrent Multipath Transfer extension of Stream Control Transport Protocol (CMT-SCTP) data transport over multiple wireless paths.

We have identified packet reordering as a major cause of performance degradation in both single and multi-path transport protocols for real time communications and media streaming. We have proposed a dynamically resizable buffer based solution to mitigate this problem within the DCCP (single path) transport protocol. For improving the performance of multipath transport protocols over dissimilar network paths, we have proposed a delay aware packet scheduling scheme, which significantly improves the performance of multimedia and bulk data transfer with CMT-SCTP in heterogeneous multi-path network scenarios.

Finally, we have developed a tool for online streaming video quality evaluation experiments, comprising a real-time cross-layer video streaming technique implemented within an open-source H.264 video encoder tool called x264.

*The only way to do great work is to love what you do.*
*If you haven't found it yet, keep looking. Don't settle.*
*As with all matters of the heart, you'll know when you find it.*
*Have the courage to follow your heart and intuition.*
*They somehow already know what you truly want to become.*
*Everything else is secondary.*

— Steve Jobs [1]

# ACKNOWLEDGMENTS

This thesis is the chronicle of my four year long journey of making various connections and interactions with research literature, successful and failed research attempts, with machines and with many wonderful inspiring people. A few words here is never going to be enough. But I would really like you all to know that I indeed feel humbled, grateful and privileged because you were with me, around me, made me feel that I belong and for all your support.

I would like to express my sincerest appreciation to Roksana Boreli and Emmnuel Lochin, my supervisors, who walked with me during this whole journey as my closest mentors. Without your support, contribution and guidance, this thesis would never have been possible. Roksana deserves my earnest gratitude for all the support, counseling and mentorship I received from her. As a student in research, I won't ever claim that I was always the easiest to deal with. Despite my efficiencies to easily get distracted among things outside the thesis and deficiencies to stay persistently motivated, Roksana truly was amazingly capable and patient to have always been there helping me reach the end. I could never have asked for better supervisors than I had.

I am grateful to NICTA and ISAE for accepting me as a cotutelle candidate and financially supporting me through this course. I have been a small part of NICTA for a very long time, since the August of 2006, and NICTA, along with my esteemed colleagues, peers and friends, has been a big part of me. I am truly indebted to all the support and encouragement I received from you.

Finally, I would like to thank my Ammu, Abbu and Yeasha, together with the rest of the family, for being so caring and supportive, for believing in me and for letting me follow what I thought one day my dreams might become. Thanks for tolerating my occasional mood-swings, prolonged inexplicable silence, countless meaningless blank stares and eternal dedication to lifeless machines. I would like you to know that I love you much more than those machines. Without you being there, I would never have accomplished this.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ALGORITHMS

# ACRONYMS

| | |
|---|---|
| ACK | Acknowledgment |
| AIMD | Additive Increase Multiplicative Decrease |
| ALF | Application Layer Framing |
| BER | Bit Error Rate |
| BGAN | Broadband Global Area Network |
| CA | Congestion Avoidance |
| CBR | Constant Bit Rate |
| CCID | Congestion Control Identifier |
| CMT | Concurrent Multi-path Transfer |
| CWND | Congestion Window |
| DCCP | Datagram Congestion Control Protocol |
| DOS | Denial of Service |
| DTX | Discontinuous Transmission |
| ECN | Explicit Congestion Notification |
| HSDPA | High-Speed Downlink Packet Access |
| IETF | Internet Engineering Task Force |
| IP | Internet Protocol |
| IPG | Inter Packet Gap |
| ITU | International Telecommunication Union |
| LAN | Local Area Network |
| LTE | Long Term Evolution |
| MOS | Mean Opinion Score |
| MPEG | Moving Picture Experts Group |
| MSS | Maximum Segment Size |
| MTU | Maximum Transmission Unit |

| | |
|---|---|
| NFB | No Feedback |
| NICTA | National Information and Communications Technologies, Australia |
| NS-2 | Network Simulator 2 |
| OSI | Open Systems Interconnection |
| PBX | Private Branch Exchange |
| PEP | Performance Enhancing Proxy |
| PSNR | Peak Signal-to-Noise Ratio |
| QoS | Quality of Service |
| RFC | Request For Comment |
| RTCP | RTP Control Protocol |
| RTO | Retransmission Timeout |
| RTP | Real-time Transport Protocol |
| RTT | Round Trip Time |
| RWND | Receiver's Window |
| SACK | Selective Acknowledgment |
| SCTP | Stream Control Transport Protocol |
| SIGTRAN | Signally Transport |
| SS | Slow Start |
| TCP | Transmission Control Protocol |
| TFRC | TCP Friendly Rate Control |
| UDP | User Datagram Protocol |
| VBR | Variable Bit Rate |
| VoIP | Voice over Internet Protocol |

# 1 INTRODUCTION

## 1.1 MOTIVATION

Real time communications and streaming media have become one of the largest, if not the main contributors to the total generated volume of traffic on the Internet. Online measurement company ComScore[1] estimates that on average, Internet users viewed 30 billion online videos per month in 2009. Cisco[3], NetworkWorld [4] and StreamingMedia [5] Internet monitoring reports on video communications also identify video streaming growth will quadruple in 2014, making video streaming one of the most pervasive applications on the Internet. Voice over IP (VoIP) has, similarly, become an integral part of our communication means, with a huge growth of traffic in developed and developing countries alike. Also with the rise of emerging new web based VoIP and video technologies such as IP based call forwarding [6, 7], Push-to-Talk [8, 9] and mobile video sharing [10, 11], real-time VoIP and video network data traffic is expected to grow even higher in the forthcoming years.

A large part of this traffic has, in recent years, been shifting to mobile or portable devices, which have also enjoyed tremendous grown in numbers globally [3, 12]. Mobile hand-held devices like smart-phones have, in recent years, become increasingly popular for a wide range of applications. These range from browsing multimedia content-rich web-pages, interactive networked multi-player gaming and VoIP telephony to streaming video playback. The huge rise of user activity on mobile devices has also proportionally increased the load and demand of mobile network capacity, and the need to use all the resources available to provide the users with the appropriate applications quality. Most modern hand-held devices today are multihomed, i.e. they are equipped with more than one network interface to ensure better network connectivity for the users. The most common network combination, 3G and Wi-Fi, is available on the vast majority of all smartphones. Therefore, wireless technology used on such devices is becoming the major component in the transmission of this increasingly important realtime communication and streaming media traffic. This includes both 3G, 4G, LTE, Wi-Fi, widely available in urban areas and, to a lesser extend but equally importantly, long range wireless e.g. satellite communications that is an integral part of the services provided in both developing countries and

[1] http://www.comscore.com

the rural remote regions of developed countries with large land masses like Australia or Canada.

A number of issues need to be addressed in order to have an acceptable service quality for real time communications in wireless environments. An additional opportunity for improving the performance of real-time streaming data services and applications, and the overall user experience in this environment is presented by the availability of multiple wireless interfaces on mobile devices. This can enable a larger available data rate and a correspondingly richer multimedia experience and audio-video quality for mobile users, assuming the available bandwidth on the multiple wireless interfaces can be aggregated and utilized for a single data flow.

In this thesis, we focus on transport protocols for real-time communications and streaming and deal with issues associated with the wireless and multi-path transport protocols. These include the Datagram Congestion Control Protocol (DCCP) [13] designed for congestion controlled data transmission over the Internet and, for multi-path data transfer, the Concurrent Multi-path (CMT) extension of Stream Control Transport Protocol (CMT-SCTP) [14]. We choose to work on DCCP as it was primarily designed for better transmission of multimedia data, while SCTP, designed to augment the fundamental limitations of Transmission Control Protocol (TCP), is a more established protocol, although the acceptance in the real world environment has been lower than the ubiquitous TCP. SCTP has gained larger support and adoption in the Telecommunications industry as Internet Engineering Task Force (IETF) defined Signalling System 7 (SS7) compatible SIGTRAN [15, 16, 17] protocol suite [18] for telecommunication signalling which takes advantage of the SCTP design benefits.

Also the recently developed WebRTC [19], real-time communication protocol standard for next generation HTTP [20, 21], has proposed the use of SCTP based data channels for better exchange and management of concurrent real-time multimedia data streams. As per the current proposal [19], multimedia data packets will use SCTP features for transport protocol (e.g. multi-streaming, congestion control, flow control etc.) and then these data packets will be encapsulated using UDP by the Datagram Transport Security Layer (DTLS) [22], before being transmitted over the link. This will ensure transport layer design benefits from SCTP while providing security features from the standardized DTLS protocol.

A major design motivation behind the proposed protocols such as DCCP [13] and SCTP [23] was to combat unfair bandwidth sharing and ever-growing Internet congestion introduced when multimedia application data are transmitted without any congestion control using the UDP protocol. DCCP along with its various pluggable congestion control algorithms and SCTP in its unreliable transmission mode, provide desired congestion control for multimedia data but no unnecessary reliability by means of packet retransmission. Unreliability with congestion control is well suited to VoIP and video applications which require timely delivery of application packets rather than reliability unbounded by any time limit.

We address the issues related to various impairments which exist in single and multi-path wireless links, including handling of errors, end-to-end delay and packet reordering. All these, combined with transport-protocol specific consequences of such impairments, contribute to lowering of user's quality of experience and a lower perception of network quality and usability. We list the specific issues we have addressed in the thesis and the contributions that improve the performance on wireless links in the following sections.

## 1.2 RESEARCH QUESTIONS

Long delay links not only directly impact the interactivity of real time applications, but can also further negatively impact their performance, by indirectly affecting the performance of transport protocols. The most common additional impact from such links is on the performance of congestion control mechanisms, as in the vast majority of the currently proposed and/or utilized transport protocols, congestion control is based on the receiver feedback received on the protocol's sender side. We first address the question of how it may be possible to improve the performance of a transport protocol designed for real time communications on long delay links, such as DCCP. DCCP congestion control mechanisms, including CCID-3 [24] and CCID-4 [25] rely on the estimation of data rate at the receiver, therefore having a long feedback delay can negatively impact the time required to detect (and subsequently react to) changes on the link, and hence on the resulting data throughput.

Most transport protocols include both congestion control and reliability components (we note that DCCP only includes congestion control and, as a design choice, excludes reliability). Congestion control mechanisms assess the congestion on the link based on the perceived packet losses at the receiver, and adjust the data transmission (rate, or window size, depending on the type of congestion control) accordingly. Packet reordering presents issues similar to those resulting from link errors, as out of order packets in most transport protocols are perceived as losses. In single path transport protocols, such as in DCCP, reordering is primarily caused by external (to the transport protocol) phenomena such as link errors, sudden route change, spurious packet retransmission etc. Therefore, as these are difficult to prevent, the question we address is how to improve the performance of DCCP in the presence of packet reordering.

The reordering problem is even more pronounced in multi-path protocols, as the imbalance in the multiple paths used for transmission has a direct impact on the level of reordering and consequently on the amount of data packets in transport protocol buffers, on the packet arrival delay, overall flow control and transmission rate of transport protocols. For multi-path data transfer, we are posing the research question of how multi-path transport protocols can fully exploit the availability of simultaneously available multiple (dissimilar) data transmission paths. This is an increasingly widespread scenario, as most current mobile devices have a combination of cellular (3G/4G) and Wi-Fi wireless technologies, which have different network characteristics and differ in the available link capacity, network delay and the level of link losses. We study the problems resulting from path imbalance and target the overall improvement in data rate over combined heterogeneous network links, with minimum required changes in the original transport protocol specifications.

Finally, mitigation of link errors originating from the wireless media is a classical research problem in transport protocols. Errors result in both (unnecessary) reduction of the perceived available bandwidth on the link, since they are treated as congestion by the transport protocols, and in increased retransmissions for fully reliable protocols. We consider the specific question of how the performance of real-time applications over multi-path transport protocols can be improved by the use of erasure codes.

## 1.3 THESIS OUTLINE

This thesis is organized as follows. In Chapter 2 we discuss the current state of the art in single path and multi-path transport protocols and relevant prior research proposals towards their enhancements. We focus on DCCP in regards to the existing problems in single path transport protocols for real-time multimedia data transmission. For multi-path data transfer, we discuss SCTP, along with its multi-path transfer extension CMT-SCTP.

In Chapter 3 we identify problems related to Voice over IP (VoIP) data transmission over DCCP, specifically on long delay wireless and satellite links. We present results from extensive evaluation of VoIP data transfer, using different voice codecs, and with DCCP over a real satellite link. We identify major performance issues in the existing DCCP protocol and its supported congestion control mechanisms, and propose solutions to mitigate them. All our proposals are validated by both simulation and experimentation on a live satellite network.

In Chapter 4 we address issues related to the presence of out of order packets in the DCCP data receiver. We propose a dynamic buffer based solution to handle packet reordering and provide extensive simulation results to support our proposal, as compared to the original DCCP protocol.

In Chapter 5 we study the performance of CMT-SCTP multi-path data transfer protocol on multiple heterogeneous data transmission paths. We identify a number of issues in regards to receiver's buffer blocking, spurious packet retransmission, and incorrect RTT estimation. We also include an analysis of receiver's buffer blocking.

In Chapter 6 we further address the issues related to multi-path transport on dissimilar network paths, focusing on packet scheduling and it's impact on the out-of-order received data packets in (single flow) multi-path data transmission using CMT-SCTP. We propose a delay aware packet scheduling algorithm to mitigate this issue and present an analysis and results of simulations to support our proposal.

In Chapter 7 we address mechanisms which aim to mitigate packet loss and receiver's buffer blocking in multi-path transport protocols and specifically CMT-SCTP, by use of erasure codes. We propose integration of erasure codes in CMT-SCTP, enhanced by the proposed packet retransmission scheme called Adaptive Retransmission (aRTX). This scheme, in combination with the erasure correction abilities of the selected erasure codes, notably improves the performance of CMT-SCTP's multipath data transfer in lossy conditions on any (or all) of the simultaneously available network paths.

Finally in Chapter 8 we discuss the tools we have developed during the course of this thesis. We first present a cross-layer video encoding and streaming tool that uses the DCCP-based transport layer for cross-layer interaction with the video application. This tool can be utilized for real-time cross-layer video streaming experiments in varying network conditions and the evaluation of video quality. We also present a second tool, called "PacketViz" which analyses, offline, packet transmission data and produces a visual representation of the life-time of each transmitted packet.

## 1.4 CONTRIBUTIONS

The contribution of this thesis address issues related to enhancing the service quality for real time communications in wireless environments, with both single-path and multi-path transport. Specifically:

- Improved quality of VoIP applications using the DCCP protocol over long delay links.

Transport protocols designed for real time communications, like DCCP, suffer a performance degradation when used over long delay links [26]. We propose a novel enhancement of the DCCP-CCID4 congestion control algorithm, that does not require any additional link rate information from the receiver (compared to standard DCCP-CCID4), and enables significant performance improvements over long delay links. We experimentally evaluate the performance of selected voice codecs which use DCCP/CCID4 with TCP Friendly Rate Control (TFRC) congestion control, over the live IPSTAR satellite network [27], 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) [28]. We evaluate the VoIP quality under different conditions of network load using the ITU E-model [29] and also evaluate fairness to competing TCP traffic sharing the same network. 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. This contribution is presented in Chapter 3.

- Improved handling of packet reordering in the DCCP protocol.

Packet reordering is a common issue (although of a varying magnitude) in various types of networks. Reliable transport protocols such as TCP inherently include a mechanism to handle out of order packet delivery, by the built-in acknowledgment and retransmission mechanism. However, due to the lack of reliability, unreliable transport protocols like DCCP lack the capability of ensuring in-order packet delivery, which has to be provided by different means. We propose a reordering-buffer based mechanism which ensures ordered packet delivery to the multimedia applications following a given time threshold. Indeed, although multimedia applications do not need reliability, their performance could significantly decrease when they do not have an in-order delivery service. Furthermore, we show that the use of a reordering buffer also improves the performance of unreliable congestion-controlled protocol such as DCCP, which interpret out of order packets as losses, indicating congestion. This contribution is detailed in Chapter 4.

- Identifying of issues and proposals for mechanisms to improve the performance of multi-path transport protocols, with application to CMT-SCTP.

The widespread of heterogeneous multiple wireless network connectivity on modern mobile computing devices necessitates changes in the well-established protocols to fully exploit this diverse networking environment. Multi-path transport protocols aim to provide the next generation data transmission over these readily available network paths, to efficiently exploit multi-homing and multi-path based parallel data transfer. With our first contribution in this area, we provide a detailed analysis of various causes of performance degradation resulting from the asymmetry of network paths in multi-path SCTP (CMT-SCTP) for reliable transmission of single data flow over multiple paths. We identify receiver window blocking, spurious retransmission, delayed selective acknowledgment (SACK) and incorrect return trip time (RTT) estimation, as the causes of the degradation. We also provide in-depth analysis of receiver's window blocking due to round-robin packet scheduling and dissimilar congestion window (CWND) growth on different paths. This contribution is described in Chapter 5.

Receiver's buffer (window) blocking is a known issue in multi-path data transfer protocols [30, 31]. Our second contribution in multi-path transport, detailed in Chapter 6, includes a proposal for a novel packet scheduling technique that exploits the awareness of per-path delay with respect to the combined overall capacity of the paths, to mitigate receiver's buffer blocking. We also propose a related technique to estimate the forward delay for delay aware scheduling and evaluate (via simulation) the performance of the proposed scheduling mechanism, as compared to the standard round robin scheduling used in current multi-path transport protocols. We additionally include a sensitivity analysis of the performance to the imperfect delay estimation. We note that, although we refer to the specific details of the CMT-SCTP protocol when evaluating the performance improvement resulting from the use of our proposed scheduling mechanism, our proposal is equally applicable to other multi-path transport protocols like e.g. MP-TCP.

Finally, we present a novel proposal eCMT-SCTP, that integrates erasure codes within the CMT-SCTP multi-path transport protocol, with regards to congestion control and reliability mechanisms. This includes a proposed enhanced packet retransmission scheme called "adaptive packet retransmission", with customizable parameters in line with the erasure code parameters. We evaluate the performance improvement compared to standard CMT-SCTP, for various erasure codes and under different levels of network losses on the multiple transport paths. Our results show that we can achieve from 10% to 80% improvements in application good-put under lossy multi-path network conditions without a significant penalty i.e. with a minimal (10%) overhead due to the encoding- decoding process. We further evaluate the performance of video streaming using an equivalent of partially reliable CMT-SCTP with erasure coding and again demonstrate a solid performance improvement for our proposal, compared to CMT-SCTP. We provide an analytical evaluation of CMT-SCTP with erasure codes and validate this with simulation results. This contribution is detailed in Chapter 7.

- Contribution to tools for video streaming

We present "Xstream-x264", a tool we have developed for *on-line* cross-layer video encoding and streaming in Chapter 8. Xstream-x264 considers input from the transport layer that uses the DCCP protocol, to enable adaptable H.264 video coding and streaming in real-time. We demonstrate its benefits to video quality on congested links through experimental evaluation.

### 1.4.1 *List of publications*

A list of publications based on the contributions of this thesis is presented below:

1. *"eCMT-SCTP: Improving Performance of Multipath SCTP with Erasure Coding over Lossy Links"*, Golam Sarwar, Pierre-Ugo Tournoux, Roksana Boreli, Emmanuel Lochin, IEEE Conference on Local Computer Networks (LCN), April 2013 [32]

2. *"Mitigating Receiver's Buffer Blocking by Delay Aware Packet Scheduling in Multipath Data Transfer"*, Golam Sarwar, Roksana Boreli, Emmanuel Lochin, Ahlem Mifdaoui, Guillaume Smith, International Workshop on Protocols and Applications with Multi-Homing Support, December 2012 [33]

3. *"Performance Evaluation of Multipath Transport Protocol in Asymmetric Heterogeneous Network Environment"*, Golam Sarwar, Roksana Boreli, Emmanuel Lochin, Ahlem Mifdaoui, International Symposium on Communications and Information Technologies (ISCIT), October 2012 [34]

4. *"On the Quality of VoIP with DCCP for Satellite Communications"*, Golam Sarwar, Roksana Boreli, Emmanuel Lochin, International Journal of Satellite Communications and Networking (IJSCN), February 2012 [35]

5. *"Mitigating the Impact of Packet Reordering to Maximize Performance of Multimedia Applications"*, Golam Sarwar, Roksana Boreli, Emmanuel Lochin, IEEE International Conference on Communications (ICC), June 2011 [36]

6. *"Xstream-x264: A Tool for Real-time H.264 Encoding and Streaming with Cross-layer Integration"*, Golam Sarwar, Roksana Boreli, Emmanuel Lochin, International Conference on Multimedia and Expo (ICME), July 2011 [37]

# 2 STATE OF THE ART

In this chapter, we present relevant related works on transport protocols and the issues associated with the research work presented in this thesis. We first present an overview of transport protocols for real time communications and media streaming, followed by background on DCCP, which is used in our single-path related research work on real time and streaming media transport. We then summarise research works related to performance improvements for (single path) transport protocols: on long delay links and in regards to packet reordering. We then present an overview of the research effort in multi-path transport protocols along with the introduction to SCTP and the multi-path version of SCTP, CMT-SCTP. We highlight prior work on the performance of multi-path transport over heterogeneous network paths, followed by a summary of research work on the use of erasure coding in transport protocols. Finally, we present a summary of cross layer solutions to improve the performane of video streaming, relevant to our work on tools contributed as part of this thesis.

## 2.1 SINGLE-PATH TRANSPORT PROTOCOLS FOR REAL-TIME COMMUNICATIONS

### 2.1.1 *Transport Protocols for Real-Time Communication*

Real-time Transport Protocol (RTP) [38] together with RTP Control Protocol (RTCP) [39] was primarily designed to support end-to-end transfer of multimedia streams over the Internet. RTP defines the standard for multimedia e.g. audio and video data packet format to transmit them over the network. While RTP defines the data format, it is the responsibility of RTCP to manage quality of service (QoS), multiple data streams and monitor the network for abrupt changes. RTP defined by the IETF motivated the technical foundations of Voice over IP (VoIP) protocols. RTP is often used along with Real-time Streaming Protocol (RTSP) [40]. RTSP is a control protocol for streaming media providing media control commands such as pause, play etc. A session in RTP is established and managed for each media stream using RTP and RTCP. An individual session can be identified by the IP addresses of the peers and the port numbers used by RTP and RTCP. It should be noted that although RTP is most commonly implemented and used in combination of User Datagram Protocol (UDP), other transport protocols such as Stream Control Transport Protocols (SCTP) which offers inherent multi-streaming support as part of the protocols and Datagram Congestion Control Protocol (DCCP) are equally capable of carrying RTP and RTCP traffic. RTP's design choice of using Application Layer Framing (ALF) [41] enables it to define multimedia data format and allow them to be carried over any other transport protocol aside UDP.

Several other unicast and multicast congestion controlled TCP friendly protocols are also proposed in the literature which attempts to improve the overall multimedia transmission and streaming performance. A detailed comparatives study of some of the protocols discussed below can be found in [42].

In [43] authors propose a loss and delay based algorithm called LDA+ which provides congestion control by gathering feedback information from RTCP. LDA+'s congestion control follows motivation from TCP's Additive Increase and Multiplicative Decrease (AIMD) behaviour. But the overall reactive nature of LDA+ is heavily feedback information based which is provided by the combination of RTCP and RTP.

In [44] authors propose another congestion control algorithm called Streaming Media Congestion Control (SMCC). SMCC works by dynamically estimating link bandwidth by using TCP Westwood [45] alike algorithm with some changes such as slow-start and forward bandwidth estimation using inter-packet arrival delay. SMCC is capable of avoiding abrupt rate changes during packet transmission, compatible to co-existing TCP flows and shares bandwidth conservatively and fairly.

Media and TCP Friendly Rate based Congestion Control (MTFRCC) [46] was proposed primarily for streamed video data transmission over the Internet. MTFRCC claims to provide even smoother rate changes than original TCP Friendly Rate Control (TFRC) [47] both in terms of short and long term averages. MTFRCC highlights the point that congestion control algorithms designed to support multimedia data in particular provides overall better quality for the users.

In Rate Adaptation Protocol (RAP) [48] authors propose a rate control algorithm for multimedia traffic using the acknowledgement data transmitted from the data receiver which helps RAP to calculate parameters such as RTT, packet loss etc. on the data sender side. Based on these parameters, particularly by using a smoothed RTT in combination with an AIMD alike rate adjustment, RAP claims to provide a much improved rate adjustment in data sending rate.

In [49] authors proposed an end to end video streaming protocol using UDP called Scalable Video Streaming Protocol (SSVP). SSVP exploits a combination of AIMD and Inter Packet Gap (IPG) to provide a smooth video transmission rate. Although using UDP as transport, SSVP introduces receiver based packet acknowledgement which helps measure network characteristics such as RTT and packet loss, using which SSVP maintains a smooth, steady and competitive transmission rate comparing to the other protocols. SSVP claims to be able to provide much better comparative video quality than the similar other protocols.

In [50] authors propose another protocol for real-time multimedia transmission called TCP Friendly Window based Congestion control (TFWC). TFWC exploits TCP like ACK mechanism and uses a TCP throughput equation based model to manage its window based congestion control, based on which it claims to provide a much fairer bandwidth sharing than the de facto TCP Friendly Rate Control (TFRC) algorithm.

In Dynamic Video Rate Control (DVRC) [51] authors propose an adaptive rate control based video streaming protocol over the Internet. DVRC takes end-users' experience together with the variation in the network condition into account to provide a required smoother rate for the video transmission. The rate control and adjustment system also considers a range of parameters from different video encoding techniques to offer an overall improved video quality for the users.

Although these protocols offer TCP friendliness, improved rate adjustment and congestion control and overall enhanced multimedia streaming quality under specific network conditions, their adoption has been very limited in the real world network scenarios, compared to protocols like DCCP and SCTP. Considering the design limitations of TCP and the completeness, modularity (e.g. pluggable congestion control algorithm in DCCP), multimedia data transmission friendliness (e.g. multi-streaming and ability to switch between reliable and unreliable transmission mode in SCTP) in this thesis we focus our research on DCCP and SCTP protocols.

### 2.1.2 *Datagram Congestion Control Protocol (DCCP)*

In order to bridge the gap between User Datagram Protocol (UDP) [52] intended for unreliable data transfer, and Transmission Control Protocol (TCP) designed as a reliable transport protocol and unfitting for real time traffic, a new transport protocol particularly for multimedia applications, Datagram Congestion Control Protocol (DCCP), was proposed by IETF [13]. Although multimedia traffic occupies a larger portion of the total Internet traffic, protocols like RTP mostly operating over UDP doesn't provide a proper mean of congestion control such that multimedia data traffic can simultaneously be fair to co-existing TCP data flows on the network link. DCCP is capable of providing congestion controlled but unreliable data delivery which TCP or UDP is incapable of. Reliability which is often provided by means of retransmission makes in time packet delivery difficult to achieve. Therefore DCCP is designed as an unreliable transport while still being congestion controlled for a better global Internet. UDP inherently designed as the opposite in nature to TCP does not provide reliability or in order data transmission, and most of all UDP doesn't provide a congestion control mechanism from the transport protocol layer and leave it as a responsibility for the application layer or the application developer. On the other hand, TCP is unable to provide unreliable unordered data delivery and enforces strict congestion controlled fair bandwidth sharing which goes against the crucial requirement of in time packet delivery for the multimedia traffic.

DCCP supports multiple pluggable congestion control algorithms known as Congestion Control Identifiers (CCID). It is left to the applications to select the appropriate CCID while establishing a DCCP connection. CCID3 [24] and CCID4 [25], which is the small packet variant of CCID3 aimed for VoIP applications, rely on TCP Friendly Rate Control (TFRC) algorithm [53] to provide congestion control in DCCP. CCID2 [54] is designed with TCP like window based congestion control. As a general rule of thumb, it is suggested that CCID3 and CCID4 should be used by applications for multimedia data such as video and VoIP, while CCID2 should be used for applications seeking maximum fair share of available bandwidth on the link. DCCP's Pluggable congestion control IDs are discussed in more details below.

#### 2.1.2.1 *CCID2*

CCID2 [54] provides TCP alike congestion control in DCCP. Congestion control in CCID2 is provided by maintaining congestion windows similar to TCP. The Data receiver using CCID2 must acknowledge packet reception since lost or damaged data packets are considered as congestion by CCID2. CCID2 takes advantage of TCP's Additive Increase and Multiplicative Decrease (AIMD) congestion control behavior. CCID2 is designed for applications to attain maximum fair share of bandwidth on the link as quickly as possible, but may not provide a smooth rate change. Therefore applications seeking a steady transmission rate shouldn't prioritize using CCID2.

#### 2.1.2.2 *CCID3*

CCID3 uses TCP Friendly Rate Control (TFRC) [53] to provide congestion control in DCCP. In an ongoing DCCP connection, RTT and loss event rate values are repeatedly estimated. CCID3 calculates the loss event rate using loss interval lengths reported in feedback packets from the data receivers. Loss event rate represents the amount of packet loss and congestion present on the network, while RTT represents the forward and reverse delay between the data sender and receiver. By using an average packet size, estimated RTT and packet loss, in combination with TCP's throughput calculation equation as shown in Equation 2.1, a target transmission rate is calculated. DCCP data sender regulates packet emission using this calculated transmission data rate. As this flow control rate is calculated using the equation from TCP's throughput model, it has been shown to be fair to the co-existing TCP flows.

$$X_{Bps} = \frac{S}{RTT \cdot \left(\mathrm{sqrt}\left(\frac{2p}{3}\right) + 12 \cdot \mathrm{sqrt}\left(\frac{3p}{8}\right) \cdot p \cdot (1 + 32p^2)\right)} \tag{2.1}$$

A data sender using CCID3 starts data transmission using slow start same as in TCP with an initial fixed rate. Later, considering the sender is still in the slow start phase, this rate is doubled in each RTT. The slow start phase exits when the sender receives a feedback from the receiver reporting a loss event. At this point, CCID3 sender enters into the Congestion Avoidance (CA) mode. RTT is estimated by the CCID3 data sender in DCCP while loss event rate is calculated by the data receiver and sent back to the sender in feedback acknowledgment packets. Since DCCP does not provide a reliable mode data transfer, in order to mitigate the impact of lost feedback packets, which carries crucial receiver side observation of the network characteristics, CCID3 makes use of DCCP'S no feedback (NFB) timer. At first, CCID3 sender initiates the no feedback timer with an expiry time of two seconds [53]. Later it is updated using the Equation 2.2.

$$\max\left(4 \cdot RTT, \frac{2 \cdot S}{X}\right) \tag{2.2}$$

In Equation 2.2, S is the average packet size and X is the target transmission rate calculated using TCP's throughput equation. Essentially it means that no feedback timer will expire at four round trip time or twice the interval between data packets, whichever is larger. When a feedback packet is lost and no feedback timer expires, CCID3 halves its sending rate. But to avoid reaching the stall mode when no packet is transmitted by the sender, the minimum sending is set to one packet every sixty four seconds. Data receiver repeatedly sends a receiver's side observation of the data received rate to the sender contained in the feedback packets. At any point of transmission, the maximum allowed value for the sender's transmission rate is limited at the twice the rate of the observed received rate by the receiver.

CCID3's design motivation involves target applications with fixed packet size and applications willing to adjust the transmission rate by changing the Inter Packet Gap (IPG) between the transmitted packets, rather than changing the packet size. CCID3 provides a much smoother rate change than CCID2 but it is not suitable for applications using variable packet size during data transfer. As CCID3 is TCP throughput equation based, it might fail to attain a fair network bandwidth share if packet size is varied during transmission [53].

#### 2.1.2.3 *CCID4*

CCID4 [55] is a congestion control type in DCCP based on the small packet variant of TFRC algorithm called TFRC-SP [56]. TFRC-SP was originally proposed to support applications demanding TFRC alike smooth rate control for small packet generating applications such as VoIP. TFRC-SP can provide equal bandwidth share for the small packet applications in environments where both small and large packet emitting applications co-exist. To prevent bandwidth starvation or demanding unfair bandwidth share among co-existing flows, TFRC-SP imposes a $10ms$ Inter Packet Gap (IPG) for small packet generating applications. Overall, the primary goal of CCID4 and TFRC-SP is to approximately achieve the same bandwidth share as a TCP flow with packets of up to the Maximum Segment Size (MSS) e.g. 1460 bytes and experiencing similar network congestion as the other flows. CCID3 and CCID4 diverges from each other in the following aspects:

MINIMUM SENDING RATE: CCID4 implements a minimum Inter Packet Gap (IPG) of $10ms$ between data packets. The specification is found in Section 4.3 of [56].

PACKET HEADER SIZE: As specified in Section 4.2 of [56], packet transmission in rate TFRC-SP is limited by a factor of the packet header size. Packet header size is not considered in calculating the rate for CCID3 applications. CCID4 rate calculation considers the packet header size for small packet applications and computes a factor called "Header Correction" factor during rate calculation.

PACKET LOSS RATE IN SHORT LOSS INTERVALS: In CCID4, in case of short loss intervals within two RTT, the loss rate is calculated by computing the actual no. of lost packets. For example, in a short loss interval such as this where N is the total number of transmitted packets and K is the number of lost packets, the loss interval will be calculated as $N/K$ as opposed to just $N$. The detailed specification is given in Section 4.4. of [56].

NOMINAL SEGMENT SIZE: In TFRC-SP, the nominal segment size used by the TCP throughput equation is considered as 1460 bytes. This is provided in Section 4.5 of [56]. There is no such specific size requirement CCID3.

### 2.1.3 *Performance of DCCP over Long Delay Links*

Researchers in [26] have extensively evaluated the performance of VoIP with CCID4 and showed that the performance of VoIP quality over long delay links deteriorates as the delay over the network link increases. In Chapter

3 of this thesis, we address the performance of VoIP with DCCP over long delay links and propose solutions to mitigate this issue.

### 2.1.4 *Packet Reordering and DCCP*

Packet reordering, a particular concern for both real-time multimedia and bulk data transfer over the Internet, has long been considered as a non pathological network behaviour [57]. More recent studies have supported this premise, for instance, the study in [58] finds that 70% of packets are correctly ordered taking into consideration one specific GEANT backbone and in [59], the authors show that 40% of the links present in their dataset effectively reorder packets. The reasons for having out of sequence packets at the data receiver side include load balancing, multiple network paths, dynamic route generation and link bonding [60]. Additionally, in mobile devices with multiple wireless network interfaces, handover between different wireless technologies may also result in out of order packets. In today's Internet, the amount of out of sequence packets is about $3\% - 5\%$ [59]. It has also been shown by previous studies that in a single stream transmission, the probability of packet reordering is increased as the transmission rate increases [60].

Transport protocols equipped with reliability mechanism, like TCP, combat packet reordering by means of retransmission. Since DCCP does not offer reliability and was designed without the capability of packet retransmission, packets arrived out of order at the receiver cause significant performance degradation in DCCP. In Chapter 4 we characterise the impact of out of order packets on DCCP perfomrance and propose a dynamic buffer based technique to mitigate this problem.

## 2.2 MULTI-PATH TRANSPORT PROTOCOLS

In this section, we first provide on overview of various multi-path transport proposals. We then summarise the main properties of the protocols chosen in our studies, SCTP and its derivative proposed for multi-path transport.

### 2.2.1 *An Overview of Multi-path Transport Protocols*

With the increased popularity of hand-held devices equipped with multiple heterogeneous radio interfaces and multi-homing capable data-centres connected to the Internet with several network access links, contemporary networking is demonstrably moving towards multi-homed and multi-path oriented communication. Yet the majority of data transfer today is still performed using traditional TCP, which neither supports multi-homing nor can provide multi-path data transfer. To address these deficiencies, a multi-path and multi-homing capable version of TCP has been proposed recently and is actively being contributed to in IETF [61].

Majority of the backbone routers in the core of the Internet today are not well capable of utilizing diverse multiple available paths simultaneously. This property of the current Internet makes it difficult to achieve multi-path networking by dynamically manipulating packet routing. In majority of the cases they only use a single path or small number of multiple paths with similar path cost [62]. Multipath TCP is the natural descendant of the original TCP because of modern applications' high demand for reliability, robustness, mobility and unpredictability (specially in wireless network scenarios) [63]. The main ideas behind multi path TCP is to incorporate multi-homing and provide applications with different downlinks through transport protocol. Multipath TCP naturally facilitates resource pooling as multiple links act as a single pooled resource. Traffic can move away from the congested link (much like in a fluid model) and larger bursts can be accommodated. Big challenges in MP-TCP are packet scheduling, congestion control and fairness. Currently there are two proposed drafts for MP-TCP:

1. TCP Extensions for Multipath Operation with Multiple Addresses [64]
2. One-ended Multipath TCP [65]

[64] adds a mechanism to TCP to explicitly add new sub-flows to an established TCP session, where each sub-flow has its own source and destination addresses. For this to work, at least one end (preferably both ends) must have at least two addresses, and both ends must implement the multipath TCP extensions. Packets are sent down different paths by addressing them to the different destination addresses available for the remote system. The multi addressed multi-path TCP has a second sequence number space carried in TCP options, so that the regular sequence number and acknowledgment fields can remain compatible with existing middle boxes such as NATs (network address translations) [62].

[65] modifies only the sender side of the protocol which means it can act like regular TCP with single source and destination. This implementation exploits the fact that congestion control is typically implemented in TCP sender. Therefore it should be possible to use MP-TCP without altering the protocol in data receiver, as long as receiver uses SACK (selective acknowledgment). An interesting future development could be - letting MP-TCP to request routers to follow selection of paths made by MP-TCP. But, since TCP option fields are not fixed in a particular location and IPv4/IPv6 fields are already in use, this is currently not being considered for near future.

Current congestion control proposals for MP-TCP includes [66]. As authors acknowledge, coupled multi-path aware congestion control specified in [66] may not be the best solution for pooling all available resources, but is completely fair to existing TCP and readily deployable.

In [67], through simulation authors demonstrate that throughput achievable through multi-path can be significantly higher than single path. A decentralized control framework of the single path can extend to the multi path in a natural manner and demonstrate a linearized stability condition.

In [68], authors propose a completely new reliable rate-based transport protocol called "R-MTP" (Reliable Multiplexing Transport Protocol) to be used in lossy wireless environments for mobile nodes. For bandwidth estimation, R-MTP depends on probing. It considers jitter and inter-arrival time into calculation of bandwidth fluctuation and for flow control. Authors claim to have an user-land implementation of R-MTP. They performed their experiments with 3 types of infrared devices (including "IRLan" and "IR-Net") and two types or wireless Ethernet for this proposal.

In [69], authors propose a new receiver based transport protocol called "RCP" (Reception Control Protocol). RCP is basically a receiver based clone of TCP. Authors argue that since modern mobile hand-held devices are equipped with multiple heterogeneous network interfaces, it is more logical to have a receiver based transport protocol over a sender based solution such as TCP. Even though RCP is based of TCP, according to the authors it provides better loss recovery, scalable congestion control and better power management. For multi-homing, RCP extends itself to Radial Reception Control Protocol ($R^2CP$). $R^2CP$ provides packet scheduling, per-path congestion control and seamless hand-off for mobile devices.

Multi-TFRC [70] is a fairly new proposal comparing to MP-TCP introduced in 2009 by Dragana Damjanovic [71] and Michael Welzl [72] which aims to address problems with multiple on going TFRC flows. It is a known problem in the network research that multiple TCP or TFRC flows will tend to occupy all available bandwidth in a link even when they are all initiated from the same source resulting in bandwidth starvation for other applications or new flows. The objective of multi TFRC is to provide an equation based weighted congestion control for multiple TFRC flows as in mulTCP [73]. It is interesting to note that Multi-TFRC and MulTCP works only as a congestion control with similar purpose while "MP-TCP" [64, 65] aims to provide a complete multi-path transport protocol having rate control , congestion control and fairness on multiple flows in consideration. Multi-TFRC working group [74] proposes for a new CCID (congestion control ID) to be defined under DCCP, since DCCP already has a fully functional implementation of TFRC in CCID3 [24] for single networked data flow. Multi-TFRC draft recommends all sub-flows should be initiated together at the same time (no dynamic flow addition or deletion) and number of sub-flows in a group of flow should not exceed 6, which appears to suboptimal limitation under current proposal. As authors suggest, number of sub-flows more than 6 will adversely affect other flows. It remains unclear how fairness is achieved under those scenarios.

pTCP [75, 76], proposes a version of parallel TCP connections where normal TCP (TCP-v) connections are managed by a striped connection manager (SM). SM manages different flows along different paths while congestion is still managed per connection (per TCP-v). According to the proposal which is supported by NS-2 simulation results, pTCP is supposed to provide bandwidth aggregation for multi-homed mobile networks and connection striping on overlay networks.

In [77], authors propose a version of TCP focusing on improving throughput and fairness for wireless or long fat network connections. They compare their implementation with application based bandwidth aggregation techniques such as GridFTP [78], TCP based DTCP [79] and TFRC based MULTFRC [80]. Authors implement different schemes for finding out optimum window size and number of connections to aggregate and demonstrates improvement with correlation based single connection (CSC) method.

In [81], authors propose a network proxy based multi-path solution for bandwidth aggregation, multi-homing, resource sharing and mobility support with reliability. It also incorporates and requires a modified version of TCP called "MTCP" to work. They include a buffer management policy to reduce packet reordering due to multi-path networking. They adopted Packet-Pair based Earliest-Delivery-Path-First (PET) algorithm for TCP applications as packet scheduling algorithm.

In [82], authors propose a scheme to modify TCP to support multiple network path or different interfaces with different IP addresses. They proposed to use TCP option field to exchange IP addresses of between the sender and the receiver. They suggest as network layer solutions will in general require changes in hardware and link layer slutions are mostly for homogeneous interfaces, transport layer solutions can be easily implemented by only changing codes/software in the operating systems. Their packet scheduling algorithm is weighted round robin and congestion is based on average RTT estimation from related interfaces.

In [83], authors propose a multi-path version of TCP based on their previous contribution Loss Tolerant TCP (LT-TCP) called "MPLOT" - Multi Path Loss Tolerant transport protocol. MPLOT incorporated forward error correction schemes (FEC). Packet scheduling for multiple network paths is done based on lower loss and shorter delay on the path.

In [84], authors demonstrate bandwidth aggregation techniques using SCTP with several modifications for load sharing and multi-homing. They demonstrate their results using simulations done with NS-2. They propose SCTP based algorithm to find bottle-necks in the network and switch network paths accordingly. They also propose packet scheduling based on RTT monitoring and apply SACK tricks to mitigate out of order packets. They propose to use SACK to notify sender of missing or out-of-order packets over high bandwidth link, while the group of packets with the missing packet still traverses over the slow link.

In [85], authors propose another version of SCTP called - "LS-SCTP", Load Sharing Stream Control Transmission Protocol, for bandwidth aggregation. LS-SCTP is capable of dynamically adding new paths during the transmission, includes path monitoring and packet assignment mechanisms. LS-SCTP separates flow control from congestion control and distributes packets based on path bandwith monitoring. LS-SCTP is also capable of handling link failure without stalling path association.

In [86], authors propose a packet scheduling technique called "Arrival Time Load Balancing" (ATLB) on top of vanilla multi-path TCP (MP-TCP, M/TCP). ATLB constantly monitors end-to-end delay including the queuing delay on the sender and chooses a path with lowest delay. According to their simulation results, authors claim that ATLB provides much better performance over conventional multi-path TCP and ensures utilization of full available bandwidth even on unstable wireless links.

In [87], author proposes a multi-path version of TCP based on SCTP called "cmpTCP". Author claims to have it implemented in a userland SCTP library. Author does not make it clear which packet scheduling algorithm was used, but it appears to be round robin by the tone of the explanation. Author proposes virtual retransmission queue on each path and used forced ACK to mitigate packet reordering. Author also demonstrates a Markov model (excluding slow-start and time-out cases) to estimate aggregated bandwidth using congestion windows on multiple paths.

In [88], authors propose another multi-path real-time best effort transport protocol focusing on video and audio transmission. According to the author, the highlights of the protocol are: (1) multi-path congestion control (2) improved QoS by path viability estimation and multi-path load balancing (3) increased end-end reliability using multiple paths. The protocol does not retransmit packet as it focuses on real-time data delivery, but provides methods to mitigate packet reordering. The strength of cmpRTCP is to transmit packets in time without having to perform any retransmission. Receiver end of the protocol imposes real-time delay tolerance limit (RTDTL) which drops packets if they are later than the RTDTL value (e. g. 50ms) which can be adjusted by the application. Author proposed to compare the results with SCTP-PR (SCTP with Partial Reliability) [89] as future work.

In [90], authors propose a generic transmission control scheme called "MTCS" (multi-path transmission control scheme). Through a mathematical model, authors figure out optimum achievable bandwidth in each path. Where the end-to-end delay is not the same, in order to mitigate the out of order packet delivery problem, authors propose packet scheduling based on calculated/probed bandwidth on each path. Authors claim that based on this bandwidth estimation approach, they highly reduce packet reordering due to multi-path packet delivery.

In [91], authors propose a generic congestion control method for multi-path environment. Authors propose to change the standard TCP/TFRC equation with a few other parameters to incorporate multiple transmission paths. Sub-flows are assigned weighted bandwidth according to these new parameters to the TCP equation.

From the long list of available candidates for the transport protocol best suited to multi-path transmission, we chose to focus on SCTP, due to it's native features such as multi-homing, multi-streaming, better security, fail-over and overall multi-path data transfer support.

#### 2.2.1.1 *Stream Control Transport Protocol (SCTP)*

Stream Control Transport Protocol (SCTP) is comparatively a new transport protocol designed to supplement the limitations of Transmission Control Protocol (TCP) with features such as inherent multi-homing, multi-streaming, reliable, unreliable and partial reliable data delivery, better security, redundancy and elimination of TCP's head-of-line blocking problem. The first SCTP RFC was published in 2000 [92] by IETF Signalling Transport (SIGTRAN) working group. It was later updated and standardized by RFC 4960 [23]. An introduction of SCTP protocol can also be found here [93].

SCTP provides unicast full-duplex data transmission between to end-points over the connectionless Internet Protocol (IP) layer in OSI network model. Due to the conservative nature of the telecommunication networks, one of the major design goals of SCTP was to provide a better transport protocol in terms of higher dependability (better than TCP) for Signalling System 7 (SS7) which is heavily utilized to transmit signalling information in the telecommunication network. SS7 requires a much more dependable network in terms of in time data delivery compared to the one provided by TCP over the IP network. SCTP is capable of providing both reliable, unreliable and partially reliable [94] mode of data transmission. Similar to TCP, reliability in SCTP is achieved by retransmission in case of lost, reordered or corrupted data. Data transmission in SCTP, alike UDP, is message oriented, whereas TCP provides byte oriented data delivery. In message oriented data delivery it is possible to mark the start and end of a data blob or message which later may be transmitted over multiple packets, but the data receiver will always receive the data blob as a whole message all together. TCP's byte oriented data delivery fails to preserve the implicit structure of the original byte stream [93]. This is often problematical for many applications since the data receiver cannot ever confidently identify if the structure of the original message from the sender changed during delivery and if the whole message is received yet. SCTP is capable of establishing association between end-points with multiple IP addresses. This allows SCTP to provide multi-homing support which TCP is incapable of providing. Multi-homing can potentially provide better fault tolerance and fail-over in case of network failures. In the next section we will find out how this feature can also be used for load balancing and throughput improvement provided by the Concurrent Multipath Transfer (CMT) [14] proposal.

SCTP carries many design similarities to TCP such as they are both connection oriented and capable of unicast data transfer. Both SCTP and TCP can provide reliable data transfer by inferring in sequence data delivery. Data integrity in TCP is provided by 16 bit Internet checksum [95] while SCTP ensures authenticity by means of 32 bit Cyclic Redundancy Checksum (CRC) [23]. SCTP is also capable of providing some UDP alike features such as unreliability and out of order data delivery. Unreliability in SCTP is achieved by disregarding retransmission of lost data chunks, while out of order delivery is realized by providing incoming data chunks immediately to the data receivers without any buffering. UDP has a packet header size of 8 bytes [52] while SCTP requires a 12 byte data chunk header [23]. Even in unreliable or unordered mode, SCTP is able to provide features such as Path MTU discovery and packet fragmentation which UDP cannot.

TCP is known to be prone to Denial of Service (DOS) attacks in the form of SYN floods [96] where the attacker tries to claim all the resources on the system by initiating numerous TCP connections just by sending SYN packets. Once TCP server consumes all the available resources on the system, it remains unavailable to newer clients or connections as no more TCP connection cannot be initiated and maintained by the system. In case of SCTP, protection from DOS is provided by the cookie mechanism SCTP uniquely introduces in its four way handshake mechanism used in connection establishment, as opposed to TCP's three way handshake. Upon receiving a connection initialization request (INIT), SCTP server generates a Cookie parameter and sends it back to the client (INIT ACK). A genuine client will receive the Cookie and will be able to return it back (COOKIE ECHO) while a blind attacker faking sender's IP address will fail to do so since he will never receive the INIT ACK data chunk and thus a successful connection will never establish in the absence of COOKIE ECHO.

TCP is prone to Head of Line Blocking problem in presence of damaged or lost packets when all the successfully received packets have to stay queued in the buffer until the missing packet is retransmitted. By the very nature, TCP infers reliable and ordered data delivery. So lost data packets causes unwanted delays during transmission. This is a critical issue for signaling networks such as SS7 where timeliness is crucial over orderliness. By means of multi-streaming and out of ordered data delivery SCTP is able to get around this inherent limitation of TCP. SCTP supports both reliable, unreliable, ordered and unordered mode of data transmission which offers an application to enjoy benefits of both TCP and UDP.

#### 2.2.1.2 *Concurrent Multipath Transfer SCTP*

Although SCTP provides in-built multi-homing, secondary paths are still only meant to provide fail-over redundancy and load-balancing. Therefore an extension to the existing SCTP standard, Concurrent Multipath Transfer SCTP (CMT-SCTP) [14], has been proposed to enable simultaneous usage of multiple available paths, aggregating the dispersed available capacity. Using SCTP's multi-homing feature, CMT-SCTP data sender distributes data over multiple physical paths in an end-to-end multi-homed SCTP association. In original SCTP proposal [23], multi-homed connections were primarily considered for fault-tolerance. But CMT extension in SCTP offers simultaneous usage of multiple physical paths in a single data flow to be utilized for an overall much higher data rate. CMT allows a single application data flow to split over multiple physical paths as sub-flows.

Although the aggregation of bandwidth through combining physical path capacity may appear simple, packet scheduling, flow and congestion control introduces several challenges in the CMT scheme such as spurious packet retransmission, conservative congestion window growth and higher acknowledgment data traffic identified in the original CMT proposal. As mitigation strategies, authors propose three algorithmic changes in the SCTP packet scheduling and acknowledgment mechanism defined in [14]. In order to minimize spurious packet retransmission, several retransmission schemes based on various transport protocol parameters such as round trip time (RTT), congestion window (CWND) and observed packet loss rate [14] are proposed. Overall, in this foundation work authors propose and demonstrate the feasibility of a novel SCTP based transport layer multi-path data transmission scheme to improve application performance through bandwidth aggregation in multi-homed scenarios.

Besides the issues pointed out in the original proposal, CMT is also known to suffer from other major problems such as receiver's window (RWND) blocking, spurious retransmission due to reordered acknowledgment (ACK) packet arrival and packet scheduling under asymmetric network path scenarios where the aggregated paths differ in network characteristics such bandwidth, delay and packet loss . We investigate into these problems in more details in Chapter 5 and propose a novel packet scheduling scheme to mitigate these issues in Chapter 6.

## 2.3 MULTI-PATH TRANSMISSION ON HETEROGENEOUS LINKS

Increasing total aggregated capacity by using the bandwidth available over multiple network interfaces in multi-homed devices, i.e. multi-path networking, for enhanced user experience is not inherently supported by traditional transport protocols. Multipath TCP (MP-TCP) [64] and Concurrent Multipath Transfer extension of the Stream Control Transport Protocol

(CMT-SCTP) [14], are the transport protocol extensions which have received most attention in the research literature to provide these capabilities. Still most research works only address an environment in which there is little difference between the multiple paths used by the transport protocol [14] or they take advantage of multiple logical SCTP streams within a single network flow [97]. This is clearly not the case with 3G (or other cellular network) and Wi-Fi on multihomed mobile devices, with the former having a lower offered bandwidth in current cellular services and significantly higher delay [98]. Although in some cases, like free public hot spots, the Wi-Fi bandwidth may actually be lower than e.g. 3G, the path asymmetry will still exist.

Receiver's buffer blocking is a known problem for both MP-TCP [99] and CMT-SCTP [100] when operating under various asymmetric network scenarios, since out of order incoming packets may occupy the entire receiver's buffer eventually stalling the whole transmission flow. Before transmitting newer data packets, both MP-TCP and CMT-SCTP's congestion control mechanism checks if the receiver's buffer has enough space by following the result given by $min(CWND_i, RWND)$ where $CWND_i$ is the data sender's congestion window for each path and $RWND$ is receiver's congestion window for the whole connection which essentially indicates the available space in the receiver's buffer. Although the potential of aggregating capacity in concurrent multi-path data transfer might seem a straightforward way to use all the available resources, out of order packet arrival leading to receiver's buffer block together with the conservative nature of congestion control may significantly degrade the expected overall performance of multi-path data transmission protocols [30, 31].

Since the original proposal of CMT-SCTP, several retransmission policy based mitigation techniques for receiver's buffer blocking have been put forward [100, 101, 102]. The fundamental idea behind buffer-block mitigation by retransmission is to provide the receiver with in order packets as fast as possible, based on different policies exploiting inherent metrics such as congestion window size, slow-start threshold, estimated path loss and round trip time (RTT), or hybrid policies using a combination of them. In [30], authors also proposed a *buffer-splitting* based mechanism which ensures reservation guarantee for in-flight packets for each path by splitting the receiver's buffer. On the other hand, present state of deployable MP-TCP still depends on increasing the buffer size estimated from the combined bandwidth delay product using the highest RTT $RTT_{max}$ of the available paths and concludes that the buffer size should be adaptable based on the type of end-point devices [99]. If communicating devices can afford to provide bigger buffers, they may achieve higher throughput. Otherwise they will receive limited throughput for the case of constrained buffer size.

In Chapter 5 we further investigate the issues related to CMT-SCTP transport on multi-path heterogeneous network links. In Chapter 6 we propose a solution to mitigate the path imbalance issues for CMT-SCTP.

## 2.4 ERASURE CODING AND TRANSPORT PROTOCOLS

Reliable transport protocols such as TCP and SCTP perform poorly in presence of lost data packets [103, 104]. Both TCP and SCTP react to any lost data packet by reducing the sender's data emission rate and by using fast recovery and time-out based retransmissions [23]. Transport layer's inability to distinguish between packet loss due to congestion in the network and lower layer data loss makes it even more difficult for the transport layer to perform well on lossy links. A potential improvement may be achieved by the use of explicit congestion notification (ECN) [105] provided by the intermediate routers, by which the congestion is explicitly indicated by including marks in the packet header. Changing conditions in mobility scenarios for mobile devices with heterogeneous wireless connectivity, including both short connectivity losses on all links due to hand-offs and erroneous wireless links, resulting in a varying magnitude of losses that could significantly impair the performance of applications, create likely scenarios for demonstrating the improvements achievable by our proposal. Although we primarily focus on the multi-path version of the SCTP protocol, CMT-SCTP [14], our proposal is equally applicable to other, both single-path and multipath transport protocols.

Now we present an overview of related work in both single and multipath transport protocols in comparison to our contributions involving integration of erasure code with CMT-SCTP transport layer in Chapter 7. In [106] Dihong et al. proposed a block code based packet error recovery scheme for SCTP called "Parity Streams" to improve performance of single and multi-stream data transfer with *single-path* SCTP. In comparison to [106], our proposal evaluates performance of not just block codes but also more error-resilient convolutional and on-the-fly erasure code for multipath CMT-SCTP protocol. In [107] Yong et al. proposed a fountain code [108] based scheme for multi-path TCP called FMTCP, capable of transmitting different encoded data blocks over different paths and is resilient to path diversity as in varying RTT, loss, jitter and capacity. In [109] Y. Hwang presented another fountain code based scheme called HMTP. In HMTP, the encoded packets over multi-homed paths are transmitted until the receiver sends an acknowledgement back to the sender. This send-until-stop transmission scheme makes HMTP to perform inefficiently and makes it prone to redundancy. In comparison to fountain code based multi-path TCP derivatives, our work complements these contributions with block, convolutional and on-the-fly erasure code adaptable to both single and multi-path based

SCTP. This send-until-stop transmission scheme results in inefficient HMTP performance with an unnecessary level of included redundancy. In comparison to fountain code based multi-path TCP based mechanisms, our work uses block, convolutional and on-the-fly erasure codes which are integrated within the multi-path based CMT-SCTP. In [110] K. K. Lam et al. evaluated and demonstrated improvement for multi-path TCP with Reed-Solomon erasure code. They concluded that using the erasure code improved TCP performance under stable channel conditions but failed to provide significant improvement under noisy scenarios. Authors concluded that the main reason for this is the use of TCP protocol, which acts as a bottleneck under noisy channel conditions. Finally, in [83] V. Sharma et al. proposed yet another modification of TCP called MPLOT to mitigate bursty and correlated packet losses under wireless mesh network environments, using erasure codes and a hybrid ARQ/FEC scheme. MPLOT aims to provide the best balance between goodput and packet delivery delay for the application layer.

In Chapter 7 we demonstrate the benefits of CMT-SCTP with erasure coding and propose an integration mechanism that enables improved performance over lossy network links, equally applicable to both single and multi-path transport protocols.

## 2.5 CROSS-LAYER SOLUTIONS FOR IMPROVED MULTIMEDIA DATA TRANSFER

There has been a body of research in cross layer approaches which have demonstrated the potential of video streaming based on cross-layer information to improve the quality of end user's viewing experience [111]-[47]. In [111], authors propose a MAC layer centric cross-layer approach to stream H.264 video efficiently over HSDPA network. Authors use JM reference software [112] for offline encoding of the video streams and NS-2 simulator [113] to simulate the network scenario. In [114], authors present another scheme of cross-layer H.264 streaming to improve performance over IEEE 802.11e based wireless networks. Authors demonstrate their ideas using offline evaluation technique through NS-2 [113]. In [115], authors present a cross-layer based method to multicast H.264 streams within a wireless WLAN. Authors use JM reference software for the encoding task. In [116], authors propose a QoS aware H.264 streaming technique over wireless mesh networks and authors adopted method of experiment is NS-2 based offline evaluation. In [117], authors use standard JM implementation to simulate their proposed cross-layer based H.264 video broadcast over wireless network. In [47], authors present another cross-layer H.264 streaming method to reduce power consumption on the associated devices. Authors use another trace driven offline evaluation tool called EvalVid-RA [118, 119], a tool-set for trace file simulation of rate adaptive MPEG-4 video, originally based on EvalVid [120] tool-set and NS-2.

As can be realized, most of these proposals contain simulated results of potential improvements, due to the lack of freely available tools which can accept cross layer information, encode video according to that input and stream the encoded video to a receiver. EvalVid [120] and H.264/AVC JM Reference Software [112] are the most commonly used freely available video encoding evaluation tools today. MPEG4IP [121] also provides similar functionality. JM software supports only H.264 reference encoding, while EvalVid and MPEG4IP support *off-line* encoding, streaming and trace-file driven analysis, requiring the user to first encode the video and in the second step, stream it. However, none of these tools provides the crucial capability of encoding and streaming at the same time and more importantly, they do not allow the encoder to adjust the bit rate and other encoding quality parameters based on external inputs in real-time. In Chapter 8 we present a real-time cross-layer *online* video encoding and streaming tool called "Xstream-x264" to fill-up these gaps.

# 3 VOIP WITH DCCP OVER LONG DELAY LINKS

In this chapter 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.

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 [122],[27], [123]. These networks have an increasing amount of multimedia and real time traffic and need to be considered in developing new protocols like DCCP. In Australia two largest satellite communications providers are Inmarsat Broadband Global Area Network (BGAN) [28] and IPSTAR satellite network [27]. IPSTAR Broadband Satellite is the world's heaviest ever civilian telecommunications satellite launched into geosynchronous orbit. IPSTAR is one of the very few satellite in orbit designed for high speed, two-way broadband communication over an IP platform and plays an important role in the broadband Internet/multimedia communication. In Figure 3.1 we show the IPSTAR satellite coverage in Australia.

The rest of the chapter is structured as follows. Section 3.1 presents an overview of the most common method for VoIP quality evaluation and summarises the common voice codec parameters. Section 3.2 provides an

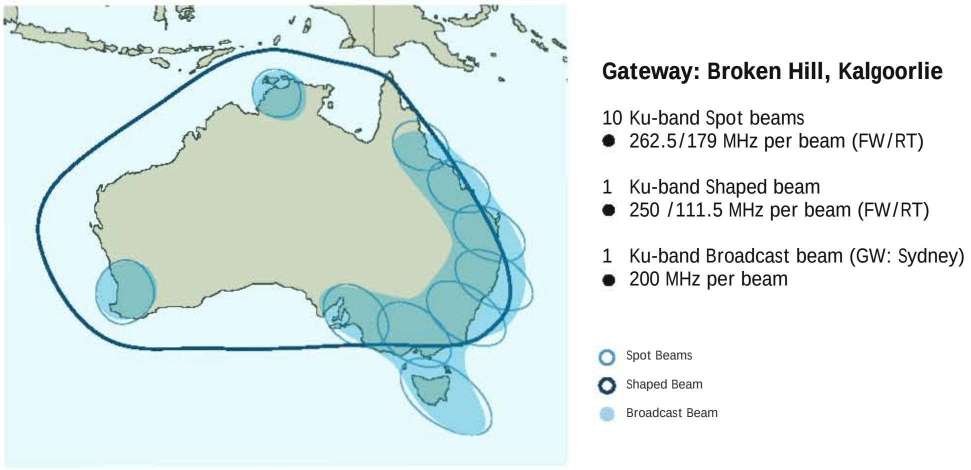

Gateway: Broken Hill, Kalgoorlie
10 Ku-band Spot beams
● 262.5/179 MHz per beam (FW/RT)
1 Ku-band Shaped beam
● 250 /111.5 MHz per beam (FW/RT)
1 Ku-band Broadcast beam (GW: Sydney)
● 200 MHz per beam
Spot Beams
Shaped Beam
Broadcast Beam

Figure 3.1: IPSTAR Coverage in Australia

overview of the related work and a description of the TFRC congestion control mechanism. Section 3.3 presents the experimental setup for live satellite tests. The following section presents initial experimental results. Section 3.5 outlines our proposed modifications to the CCID4 protocol, followed by the evaluation of VoIP quality for all experimental scenarios in Section 3.6. Section 3.7 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 Section3.8 and a discussion of the advantages and disadvantages of using DCCP for VoIP in Section 3.9. Finally Section 3.10 concludes and outlines plans for future work.

## 3.1 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 3.1 lists details of codecs commonly used in IP telephony, G.711 [124], G.729 [125] and Speex [126].

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

| | 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 |

Table 3.1: Voice codec parameters

protocol, the DCCP protocol evaluated in this chapter or other protocols which may be used, e.g. RTP [127][128]. 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 [129]. 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 [29], 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 [130]. ITU recommendation G.1020 also defines the VoIP gateway-specific reference points and performance parameters [131].

Other methods for evaluating speech quality include Perceptual Evaluation of Speech Quality (PESQ) [132]. PESQ essentially models Mean Opinion Score (MOS) results over a scale of 1 to 5 representing bad to excellent voice quality. PESQ and MOS results are interoperable and there is an ITU proposed mapping function defined in [133] which converts MOS results to PESQ, and vice versa. However, in order to remain coherent and comparable to the previous work presented in this thesis, we have only considered MOS in our work.

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

$$R = R_0 - I_s - I_d - I_e + A \tag{3.1}$$

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 [134], 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 chapter 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 [26] 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:

$$MOS = 1 + 0.035R + 7.10^{-6}R(R - 60)(100 - R) \tag{3.2}$$

[29, 129]

The quality levels defined by ITU [29] relate to satisfaction of users conducting the telephony (or VoIP) conversation and are reproduced in Table 3.2.

| R value<br>(lower limit) | MOS value<br>(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 |

Table 3.2: Provisional guide for the relation between $R - Score$ and user satisfaction

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

## 3.2 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 [24] and DCCP/CCID4 [25] use TFRC [135], [53] 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 [24]. After the receiver feedback is available, the sender's initial rate is calculated as per equation (3.3):

$$X = \frac{min(4 \cdot s, max(2 \cdot s, 4380))}{RTT} \tag{3.3}$$

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 (3.4):

$$X = min(2 \cdot X, 2 \cdot X_{recv}) \tag{3.4}$$

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 (3.5) approximating the transmitted rate to what would be an equivalent rate of TCP under the same network conditions.

$$X = s \cdot f(p, RTT) \tag{3.5}$$

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

Where: p is the loss event rate.

RTO is the TCP retransmission timeout value in seconds.

CCID4 [25] differs from CCID3 only in the congestion avoidance phase. To calculate the sending rate X, in place of the packet size s in equation 3.5, CCID4 uses a fixed packet size of 1460 bytes modified by a header correction factor, according to the following equation (3.6):

$$X = 1460 \cdot \frac{s}{s + oh} \cdot f(p, RTT) \tag{3.6}$$

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 [53] 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 [136], [137] using simulation, and over generic links in [26] using emulation. The authors propose the use of Quick-Start [138] and Faster Restart [139] mechanisms and show that these methods provide only a partial improvement to the DCCP performance over a long delay network. In this chapter, 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 [140]. 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 chapter, 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.

## 3.3 EXPERIMENTAL SETUP

Our experimental setup at the NICTA Laboratory in Sydney, Australia is presented in Figure 3.2. 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 its 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 3.5.

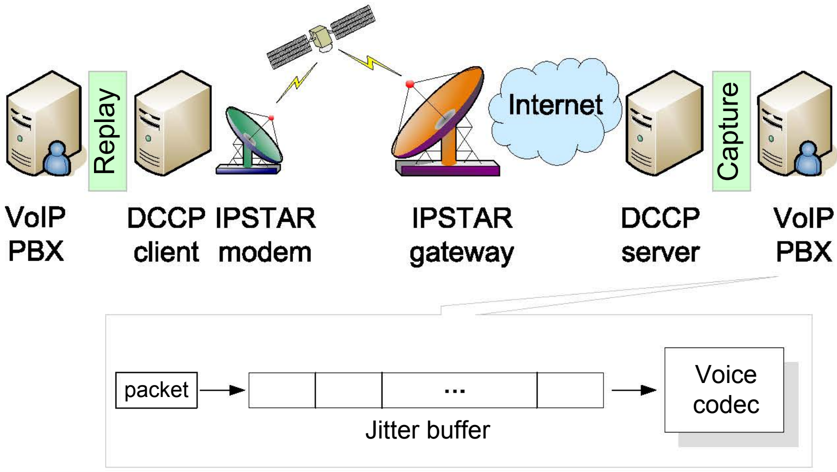


Figure 3.2: Experimental setup for live tests

The VoIP application used is Asterisk Private Branch Exchange (PBX) [141], with voice codecs commonly used in IP telephony. We use G.711 [124], Speex [126], with and without discontinuous transmission (DTX) and G.729 [125].

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 [141] 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 [140] 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 $10^{-7}$ [142].

Inmarsat BGAN also uses shared bearers, with a nominal data rate of 492kbit/sec on both downlink and uplink [123]. Access control is implemented using TDM/TDMA [143]. 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 [144], 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.

## 3.4 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 3.6. 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 3.3.

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 3.4, 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 3.1, 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 [142]. Our observations show that the main source of jitter is the network rather than the DCCP conges-

| Voice Codec<br>& load | Data rate<br>(kbit/s) | CCID4 (%)<br>(%) | UDP<br>(%) |
|---|---|---|---|
| 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 |

Table 3.3: Average packet loss rate values (%) for different codecs measured on IPSTAR link

tion 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 3.6.

| | 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 |

Table 3.4: Jitter values in milliseconds for all codecs

### 3.4.1 *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 3.1). 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 3.3 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 3.3 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 (3.6) 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 [145] with different values of RTT. We use the IPSTAR equivalent data rate, as presented in Section 3.3, and there are no losses on the link. Figure 3.3 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.

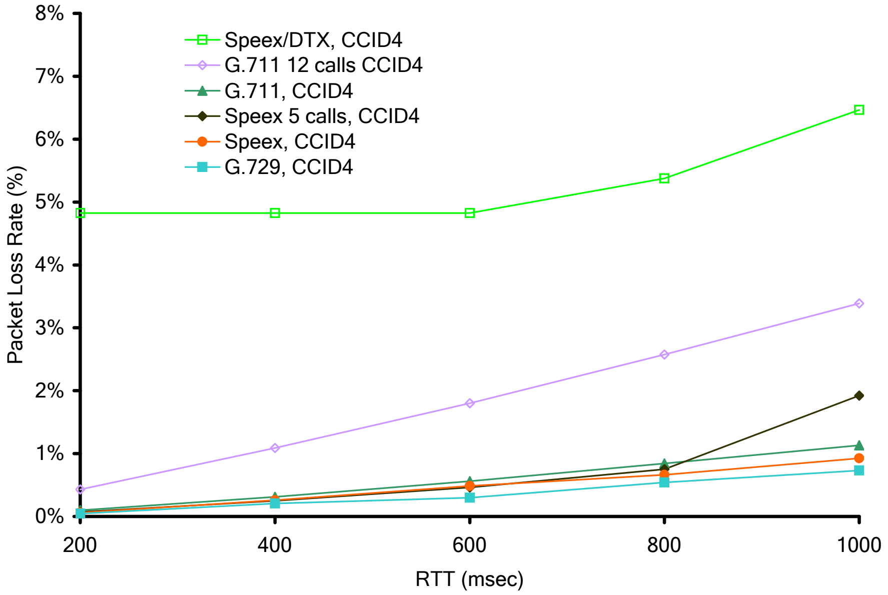


Figure 3.3: Packet loss rate for increasing RTT values, emulator, voice codecs: G.711, one and 12 calls; G.729; Speex, one and five calls

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.

## 3.5 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.

### 3.5.1 *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 (3.3) in rate calculations:

$$X = \frac{4380}{RTT} \cdot \frac{s}{s + oh} \tag{3.7}$$

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

### 3.5.2 *Increasing the Number of Feedback Messages*

We also apply the CCID3 modification proposed in [140], 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 (3.4) with increased frequency and, during the congestion control phase, provides more accurate values for changes of the RTT parameter used in formula (4.1) 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 [146].

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 [53].

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:

$$N_{Fb} = \min\left(\max\left(\text{round}\left(\frac{RTT}{RTT_{ref}}\right), 1\right), \frac{RTT * X_{rec}}{s}\right) \tag{3.8}$$

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 (3.8).

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:

$$D_{min} = RTT + O + t_{delay} \tag{3.9}$$

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 [135]. 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:

$$D_{min} = \frac{N_{Fb} + 1}{N_{Fb}} * RTT + t_{delay} \tag{3.10}$$

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:

$$X_t = s * 2^{\frac{t}{2*RTT}} \tag{3.11}$$

with standard rate control and

$$X_t = s * 2^{\frac{t*N_{Fb}}{(N_{Fb}+1)*RTT}} \tag{3.12}$$

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 Equation 3.5, 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 (3.5) during the periods with no errors, as RTT in these periods varies more than p.

### 3.5.3 *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 (3.7), the calculation of the rate in the slow-start phase becomes:

$$X = 43800 \cdot \frac{s}{s + oh} \tag{3.13}$$

The rationale of this idea is similar to TCP-Hybla [147] 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 (3.4).

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.

## 3.6 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 3.1, for a specific jitter buffer duration.

Table 3.5 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 3.3.

| | 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 |

Table 3.5: Average packet loss rate values (%) for different codecs measured in experiments on IPSTAR link

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 3.6 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 3.4. 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.

| | 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 |

Table 3.6: Jitter values in milliseconds for all codecs

### 3.6.1 *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 [148]. 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. 3.4 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 [135] are still followed.

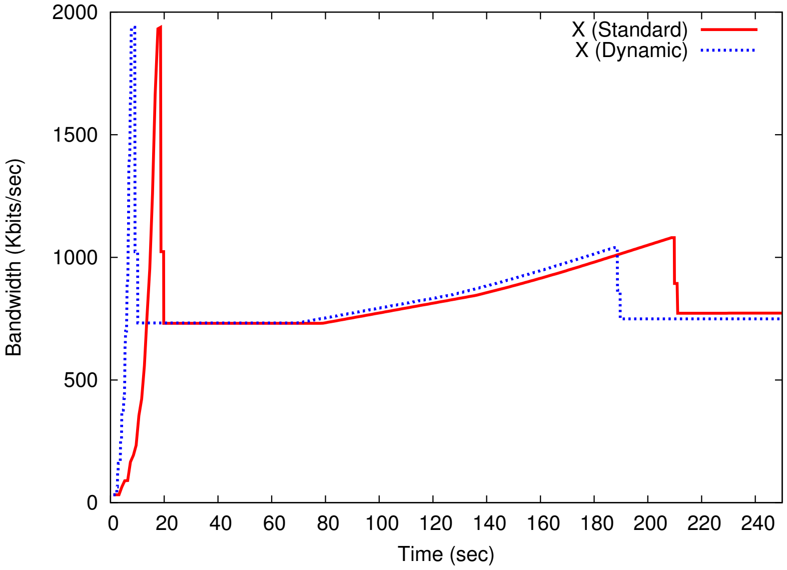


Figure 3.4: Slow start and error recovery on a representative satellite link, ns-2 simulation, standard and dynamic feedback DCCP/CCID3

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

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

Table 3.7: DCCP download performance on IPSTAR

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 is presented in Table 3.7, 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).

### 3.6.2 *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 3.2. 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 [29]. However, the maximum acceptable delay for toll quality satellite links is 400msec [149] [150], 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 [151], 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 3.5 shows the $R - Score$ factor for selected experimental scenarios on the IPSTAR satellite network for all the transport protocols considered.

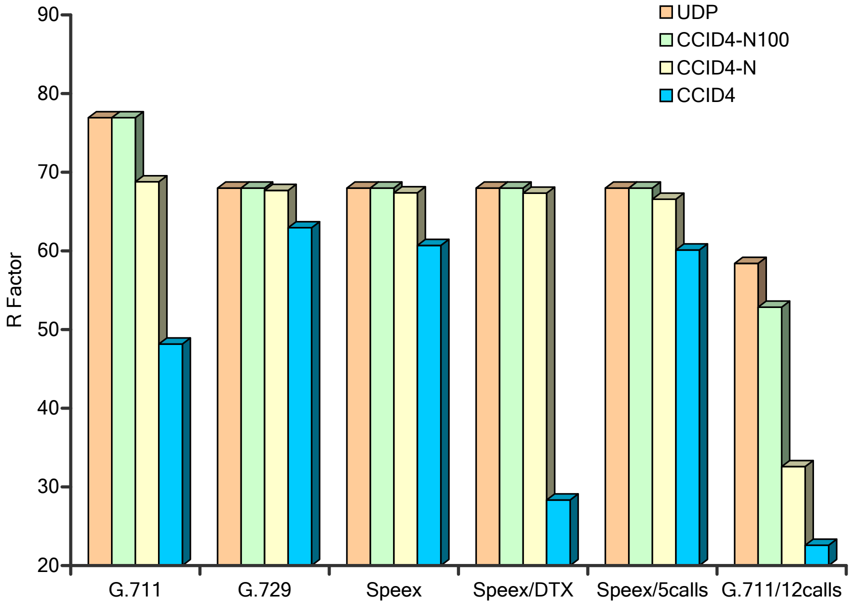


Figure 3.5: $R - Score$ factor for IPSTAR experiments, G.711, one and 12 calls, G.729, Speex, 1 and 5 calls, UDP, CCID4 and CCID4-N

To provide a clear view of the difference between the voice quality with UDP and with CCID4 variants, Figure 3.6 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.

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

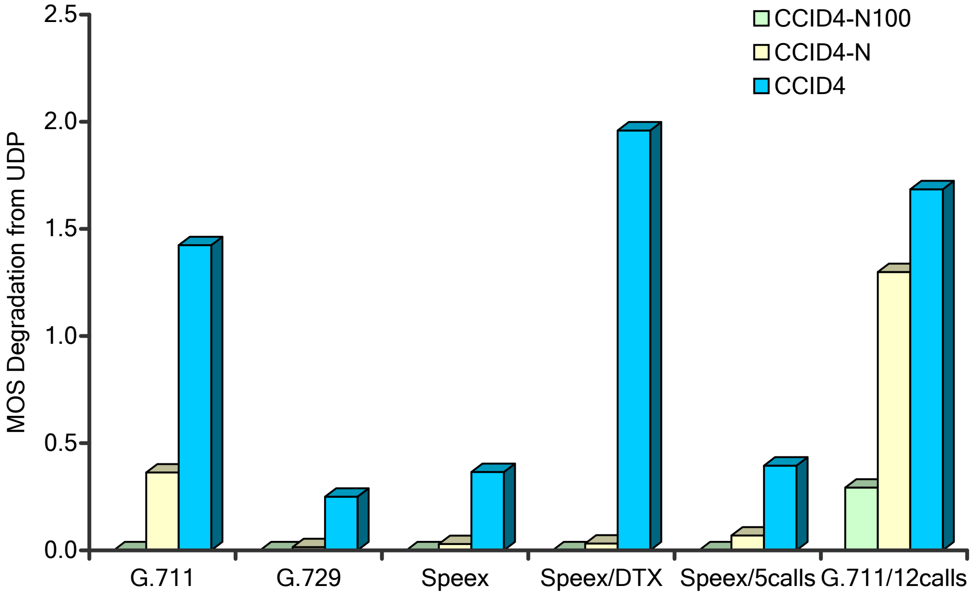


Figure 3.6: Degradation of MOS values from UDP for different congestion control mechanisms on the IPSTAR network

## 3.7 FAIRNESS TO OTHER FLOWS

To evaluate fairness of multiple flows, we use Jain's fairness index [152]. Raj Jain's fairness index is represented by Equation 3.14, where $n$ is the total number of concurrent flows and $x_i$ is the bandwidth allocated for the $i_{th}$ flow. The output of Jain's index ranges from $\frac{1}{n}$ to 1 where $\frac{1}{n}$ represents the worst fairness allocation and 1 represents the best possible fairness distribution across the corresponding flows.

$$f(x_1, x_2, x_3, ..., x_n) = \frac{(\sum_{i=1}^{n} x_i)}{n. \sum_{i=1}^{n} x_i^2} \tag{3.14}$$

| Fairness Index | CCID4 | CCID4-N | CCID4-N100 | UDP |
|---|---|---|---|---|
| G.711/12calls | 0.9997 | 0.99997 | 0.99997 | 0.74 |
| 1Mbit/s data rate | 0.985 | 0.98 | 0.98 | 0.5 |

Table 3.8: Fairness index values for two flows, TCP and CCID4 versions

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 3.8.

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.

## 3.8 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 [53], 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 [153], 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 [2] analysed the benefit brought by RFC 5348, as compared to RFC 3448. Figure 3.7 (by courtesy of the authors of [2]) 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.

| | 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 $X = \mathrm{max}(\mathrm{min}(X_{calc}, 2 \cdot X_{recv}), \frac{s}{t_{mbi}})$ | Not limited by receiver rate but by an average of the two last values of recent $X_{recv}$ values contained in $X_{recv\_set}$ in $X = \mathrm{max}(\mathrm{min}(X_{calc}, recv_{limit}), \frac{s}{t_{mbi}})$ with $recv_{limit} = 2 \cdot \mathrm{max}(X_{recv\_set})$ depending on whether the feedback packet reports a new loss event or an increase in the loss event rate |

Table 3.9: Difference between RFC 3448 and RFC 5348

As shown in Table 3.9, 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 3.7 (by courtesy of the authors of [2]) 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 3.9) 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 [139] 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 [53]. Indeed, the simulation scenario in Figure 3.7 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 [154]. 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.

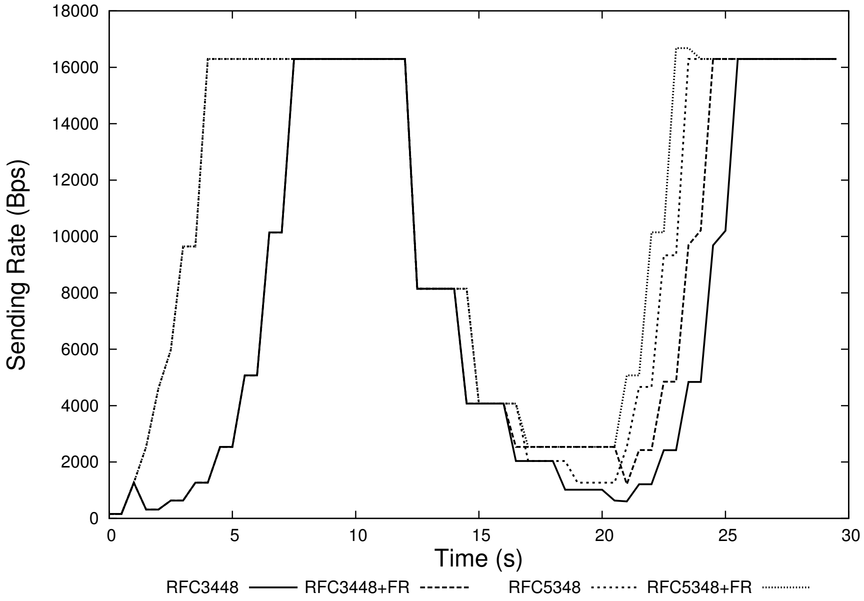


Figure 3.7: Sending rate dynamics of RFC 5348 and RFC 3448 with and without FR (by courtesy of the authors of [2])

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 [13] 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 [155]. 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 [2]).

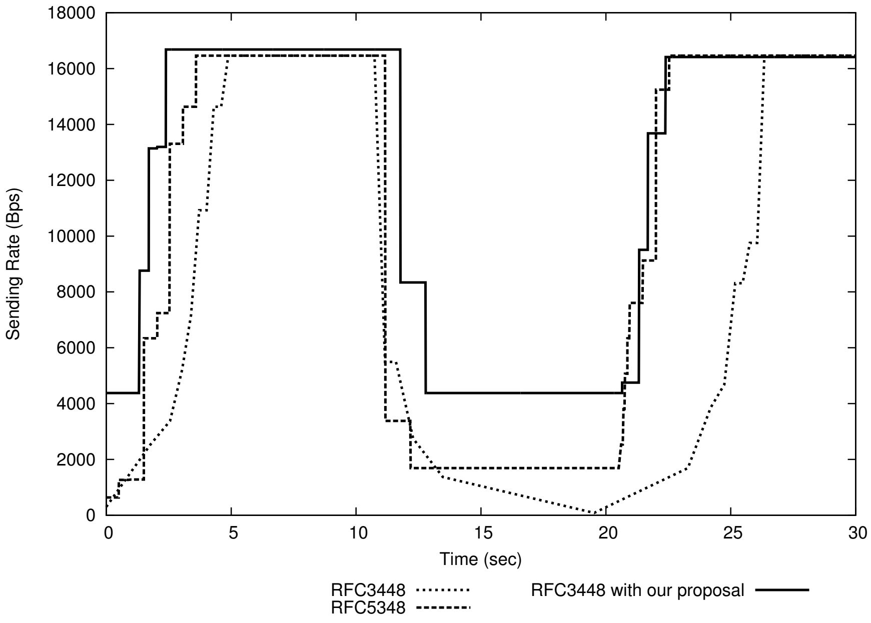


Figure 3.8: Sending rate dynamics of RFC 3448 and our proposal

Finally, Figure 3.8 presents the outcome of a similar experimental scenario as what was used to produce results shown in Figure 3.7. We reproduce the experiment using the NS-2 code for RFC 5348 of the authors of [2]. It can be clearly observed that our proposal does not contradict with the outcome of RFC 5348. As shown in Figure 3.7 and 3.8, 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.

## 3.9 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 [156]. 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.

## 3.10 CONCLUSION

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 chapter, 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.

# 4 MITIGATING PACKET REORDERING IN DCCP

In this chapter we propose a solution to mitigate the performance degradation and corresponding Quality of Experience (QoE) reduction caused by packet reordering for multimedia applications which utilise unreliable transport protocols like the Datagram Congestion Control Protocol (DCCP). We analytically derive the optimum buffer size based on the applications data rate and the maximum delay tolerated by the multimedia application. We propose a dynamically adjustable buffer in the transport protocol receiver which uses this optimum buffer size. We demonstrate, via simulation results, that our solution reduces the packet loss rate, increases the perceived bandwidth and does not increase jitter in the received applications packets while still being within the application's delay limits, therefore resulting in an increased QoE of multimedia applications. The rest of this chapter is organised as follows: Section 4.1 outlines our proposal, Section 4.2 presents the details of the simulation setup, parameters and results, Section 4.3 shows the improvements achieved in the QoE for example VoIP and video applications and we conclude in Section 4.4.

## 4.1 DYNAMIC RECEIVER BUFFER

We propose to use a receiver buffer with a dynamically adjustable size to handle the out of order received packets. We first present a rationale for our proposal, followed by the details of buffer data management. Our aim is to minimize the residual negative effects which such a buffer could have on the multimedia applications quality, i.e. the increased delay and jitter.

The proposal focuses on DCCP [13] protocol, which includes a number of options to enable the type of congestion control suited to applications requirements. As our interest is in VoIP and multimedia applications, we choose TFRC based DCCP Congestion Control IDs CCID3 [24] and CCID4 [55] to evaluate the effects of reordering. TFRC defines a rate based congestion control mechanism. After the initial slow-start like period, the sender will regulate the transmitted rate based on the receiver feedback. The receiver considers the received packet sequence numbers to detect losses and estimate the packet loss rate, indicated by the loss event rate p. This, together with the estimate of delay and the received data rate, is included in feedback packets to the DCCP sender. The sender adds the estimated receiver to sender delay to derive the return trip time and calculates the data rate for the latest reporting period by using a combination of the receiver rate and a rate calculated by the equation (4.1). This approach is used in order to provide fairness to TCP flows on the same link.

$$X = \frac{s}{RTT \cdot \sqrt{\frac{p \cdot 2}{3}} + RTO \cdot \sqrt{\frac{p \cdot 27}{8}} \cdot p \cdot (1 + 32 \cdot p^2)} \tag{4.1}$$

where: $s$ is the packet size in bytes; $p$ is the loss event rate; $RTO$ is the TCP retransmission timeout value in seconds.

The generic TFRC mechanism described above is used in the CCID3 type of congestion control. CCID4 [55] differs from CCID3 in that it is adjusted to small packets sizes appropriate to VoIP. In place of the actual packet size, CCID4 uses a fixed packet size of 1460 bytes modified by a header correction factor. This ensures that the formula based rate from equation (4.1) which is directly proportional to the applications packet size, does not unfairly disadvantage DCCP, by using a common TCP packet size rather than the smaller size VoIP packets.

It is worthwhile to highlight the effect of out or order packets on both the transport protocol mechanisms and to the applications and some differences between how TCP, UDP and DCCP handle these packets:

1. When the TCP retransmission timer (RTO) expires, TCP triggers a Go-Back-N recovery procedure which could lead to retransmissions of packets effectively received (i.e. spurious retransmissions). This case can occur when TCP considers losses following the reordering of network packets or a significant increase of the RTT (e.g. in case of a vertical hand-off, load balancing, or networks misconfiguration). In this particular case, known as spurious timeout, the acknowledgment packets get back too late to reset the retransmission timer. These indications of false losses strongly impact on the overall performances of TCP in terms of reduced achievable throughput;

2. UDP has no notion of sequence numbers and out of order packets are forwarded to the application which, assuming it has a buffer (which is the case for all VoIP and most video applications), could reorder and utilize the packets;

3. DCCP's congestion control mechanism treats reordered packets as losses, although, similarly to UDP, the out of order packets are still delivered and may be of use to the application. However, the packets following the falsely detected congestion event will arrive at a lower rate (similarly to TCP). This follows the equation (4.1) driven sender rate and, depending on the level of reordering, could present a serious issue particularly for video applications. The artificially lowered rate

could lead to losses between the sender side application and transport, which may not be able to handle the rate offered by the application.

In order to illustrate the impact of network reordering on DCCP/CCID3-CCID4, we drive a simple experiment where we simulate the network reordering effect by desequencing a number of selected packets with Linux NetEm [1]. In this experiment, we use a single bottleneck link of 1Mbit/s and uniformly distributed 5% random packets are delayed by a specified amount of time to create out of order packets (see NetEm parameters in the second column Table 4.1). No losses (i.e. due to error link) are artificially introduced during the experiment. As shown in Table 4.1, the higher the network reordering ratio, the higher the loss event rate (LER). As is to be expected based on equation (4.1), this increase leads to a decrease of the DCCP throughput (for comparison purpose the last column of the table gives the throughput obtained no network reordering is introduced).

| RTT | NetEm Reordering<br>Parameters | X<br>in Kbit/s | LER | X without<br>network reodering |
|---|---|---|---|---|
| 50ms | 25ms +/- 8 | 935.3 | 1.66 | 961.3 |
| 100ms | 50ms +/- 20 | 565.1 | 1.79 | 959.1 |
| 200ms | 100ms +/- 40 | 500.2 | 1.83 | 954.5 |
| 300ms | 150ms +/- 80 | 153.4 | 2.45 | 936.8 |
| 400ms | 200ms +/- 160 | 90.6 | 3.11 | 941.2 |

Table 4.1: DCCP bit-rate (X) for varying RTT and reordering rate

The reduced performance of both transport and applications supports the validity of the idea of introducing a receiver side buffer specifically for handling out of order packets. However, a simplistic addition of a receiver buffer has a potential to increase the delay and jitter of the received data stream, therefore negatively impacting the applications QoE, as will be shown in Section 4.3. We aim to minimize the QoE degradation by optimizing the buffer size to suit applications characteristics.

### 4.1.1 *Sizing the Reorder Buffer with Network Calculus*

We propose to implement a buffer to reorder packets before delivering them to the application. Therefore, we need to size this buffer as a function of the application constraints. Indeed, contrary to the elastic applications using the in-order delivery service provided by TCP, multimedia applications are characterized by strong delay bounds. For example, VoIP applications are

[1] http://www.linuxfoundation.org/en/Net:Netem

characterized by a maximum "mouth-to-ear" threshold where a delay lower than around 200ms [29] is necessary to get a good communication quality, while video-conferencing is known to perform ideally when the delay does not exceed 100ms [157]. We thus propose to assess the size of this buffer following a given delay threshold provided by the application denoted $D_{max}$.

We denote $R(t)$ the rate at which data exits the IP level and enters the re-ordering buffer and $R^*(t)$ the output rate of the re-ordering buffer. When the re-order buffer is disabled: $R(t) = R^*(t)$.

In Fig. 4.1, $L$ is corresponding to the maximum burst size and $R(t) = \lambda \cdot t + L$. We assume that $D_{max}$ , is the maximum delay tolerated by the application and $d(t)$ be the delay induced by the reordering queue ($d(t)$ might be considered as negligible as we only consider small buffer sizes). For the sake of simplicity, in Figure 4.1 $L = 1$ packet and $R(t) = 1$ pkt/s.

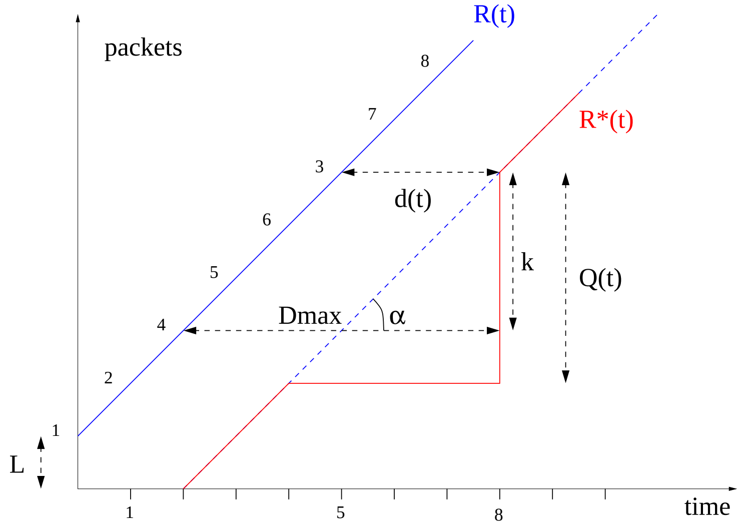


Figure 4.1: Arrivals curves $R(t)$ and $R^*(t)$

As shown, packets 1 and 2 arrive in-order and are transmitted to the application. Then, a rupture in the sequence numbers occurs when packet 4 arrives. Packet 4 and the following are enqueued in the reordering buffer until the reception of packet 3. When packet 3 is received, the reordering buffer is flushed without delay (this explains why $R^*(t)$ is perpendicular to the x-axis at $t = 8$).

PROBLEM: *knowing* $R(t)$, *what is the maximum value of the reordering buffer* $Q(t)$ *constrained by* $D_{max}$ *?*

Following Fig. 4.1, we have $\tan(\alpha) = \frac{k}{D_{max} - d(t)}$ and $Q(t) = k + L$, then:

$$\begin{aligned} Q(t) &= \tan(\alpha) \cdot [D_{max} - d(t)] + L \\ &= \lambda \cdot \left[ D_{max} - \frac{L}{\lambda} \right] + L \\ &= \lambda \cdot D_{max} \end{aligned}$$

As a matter of fact, $\lambda$ is equal to $X$ when DCCP-CCID3 is used, then $Q(t)$ becomes:

$$Q(t) = X \cdot D_{max} \tag{4.2}$$

In other words, for a given $D_{max}$, the maximum buffer size at instant $t$ is given by the product of $X$ and $D_{max}$. It also means that for a multimedia application characterized by a peak rate $R_{peak}$, we can size at the beginning of the connection $Q(t)$ to $R_{peak} \cdot D_{max}$.

### 4.1.2 *Reorder Buffer for Unreliable Transport Protocols*

Our proposal is in many ways similar to a flexible buffer common in VoIP and streaming video applications. It is applicable to DCCP or any other unreliable transport protocol which has the capability to detect the order of received data packets using e.g. sequence numbers.

Multimedia packets, on being received by the transport protocol receiving side, are, in the early stages when the initial applications data is being received, directed to a receiver buffer rather than passed on to the application. The buffer is filled up to the current maximum buffer size defined by equation (4.2) before the forwarding of data frames to the application begins.

The buffer is kept full at all times, and the received packets are accessed in regular intervals, as determined by the standard DCCP protocol. The buffer is scanned for the next packet, according to the required sequence number and forwarded from the buffer to the DCCP mechanism and the application.

Every incoming packet is assigned a relative position in the queue based on the maximum and minimum sequence number of existing packets in the buffer. Out of order packets based on their early or late arrival fill up the respective gaps in the queue. Dequeuing process simply selects a packet from the head of the queue which is the most relevant in order packet for the application. If an out of order packet which corresponds to the next-in-sequence packet is not available in the buffer at the time it is expected by the DCCP mechanism, it is considered lost and will contribute to the DCCP loss event rate.

## 4.2 VERIFICATION AND ANALYSIS

We have implemented our proposal within the DCCP transport protocol in the NS2 simulator [113]. In all the simulations we use a simple dumbbell topology, with the DCCP sender and receiver connected with two routers which introduce reordering in the data packet stream. The reordering is achieved in the same way as described in the example shown in Table 4.1. We vary the amount of delay on randomly selected packets, based on a uniform distribution, to create reordered packets. Using a uniform distribution is considered sufficient for the purpose of the initial evaluation, although we plan to consider other distributions in future work. We vary the applications data rate and RTT values and the percentage of reordered packets.

Our initial ns-2 implementation includes a fixed size buffer which we choose in the following way. For a video application, we assume the overall acceptable delay to be 100msec, consistent with [157] and a commonly used constant video rate of 1Mbit/sec. For the VoIP application, we choose two commonly used voice codecs, G.711 and G.729, with corresponding data rates (including IP and transport protocol overhead) or 80kbit/sec and 24kbit/sec. The QoE for VoIP applications [29] indicates the critical one way delay to be around 200ms, which if increased will more significantly impact the quality compared to delays below that value. As introducing the buffer will increase the already existing delay on the route between the sender and receiver, we allow for the buffer to contribute a $D_{max}$ of 100msec to voice delay and 50msec to video delay. We note the most common delay on the Internet being around 50 msec [146]. Considering the average packet sizes of 500bytes for a H.264 video packet [157], 200 bytes for G.711 and 60 bytes for G.729, we derive the maximum buffer size of 12 packets for video and 5 packets for VoIP.

### 4.2.1 *Simulation Results*

We perform a series of simulations to demonstrate the benefits of our proposal. We vary the reordering rate, RTT and buffer size and record the relevant DCCP parameters and the receiver side jitter. On the DCCP receiver side we also monitor the received rate and loss event rate and on the sender side we monitor the DCCP perceived RTT values. To enable thorough experiments, we have modified the NS-2 implementation [158] of the TCP test suite defined in [159] to handle DCCP with and without a reordering buffer. TCP specific parameters of course could not be monitored, however most of the generic parameters (e.g. RTT and throughput) have equivalent values for DCCP, although they may be derived in a different way.

To highlight the difference between the simulation results obtained using configurations with and without the reordering buffer, we define the efficiency ratio $\varepsilon$ for a parameter $z$ as follows:

$$\varepsilon(z) = \frac{z_b}{z_{nb} + z_b}$$

where $z_b$ is the parameter value obtained by simulations which include the reordering buffer and $z_{nb}$ is the corresponding parameter value resulting from simulations with no reordering buffer. For all experiments, when :

- $\varepsilon(z) = 0.5$: both schemes obtain the same performance;
- $\varepsilon(z) > 0.5$: the values obtained by the reordering buffer are higher than without reordering buffer;
- $\varepsilon(z) < 0.5$: the values obtained by the reordering buffer are lower than without reordering buffer.

We simulate an ideal application which will use the available data rate i.e. transmit the maximum amount of data using DCCP/CCID3 on a 1Mbit/sec bandwidth limited link. We present a subset of the derived efficiency ratio results for the same range of RTT and reorder buffer values, but choosing one reordering rate of 5% consistent with [59]. Figure 4.2 shows the resulting efficiency ratio for throughput, Figure 4.3 for the DCCP loss event rate and Figure 4.4 for the receiver side jitter.

The jitter, as shown Figure 4.4, obtained by both schemes is similar, i.e. the reordering buffer does not impact the jitter values in the received multimedia stream. We can observe that the losses are lower with the reordering buffer (see Figure 4.3). As a consequence, the resulting performance in terms of throughput is improved (see Figure 4.2). The throughput improvements range from small for a buffer size of 2 and low RTT (100-150msec) to a substantial 0.85 for a buffer size of 7. It is interesting to observe how the increased RTT reduces the reactivity of DCCP and amplifies the negative consequence of the reordered packets on the DCCP rate control mechanism. We note, as per the presented analysis of potential maximum buffer length values, that all the simulated buffer size values are lower than the estimated maximum buffer size of 12 packets for video. Additionally, that for a buffer size of 5 packets which has been considered the maximum for VoIP, we have an improvement (a throughput efficiency ratio of 0.6) even for the lowest simulated RTT value of 100msec.

## 4.3 APPLICATIONS QUALITY IMPROVEMENTS

QoE for multimedia applications is most commonly represented by the value of the Mean Opinion Score (MOS). In changing network conditions,

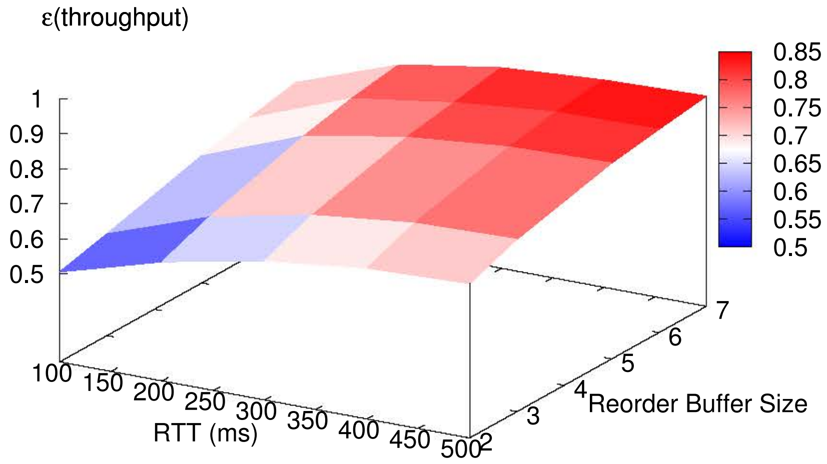


Figure 4.2: Throughput efficiency ratio

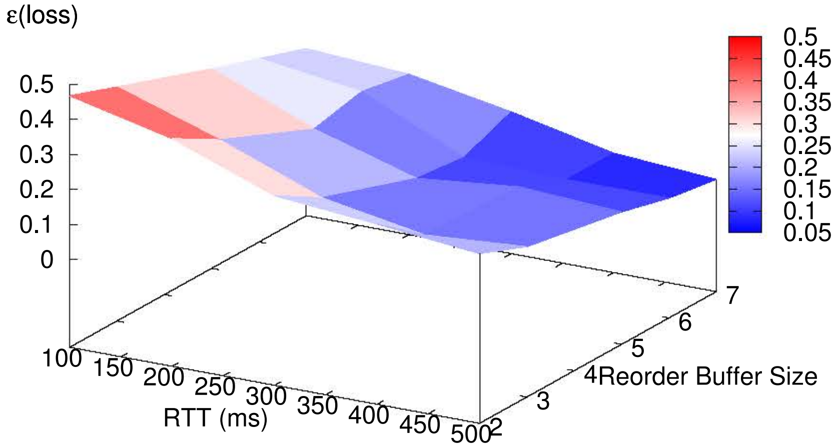


Figure 4.3: Loss event rate efficiency ratio

MOS is derived using an objective evaluation methodology, the E-Model, defined both for voice [29] and video [160] applications.

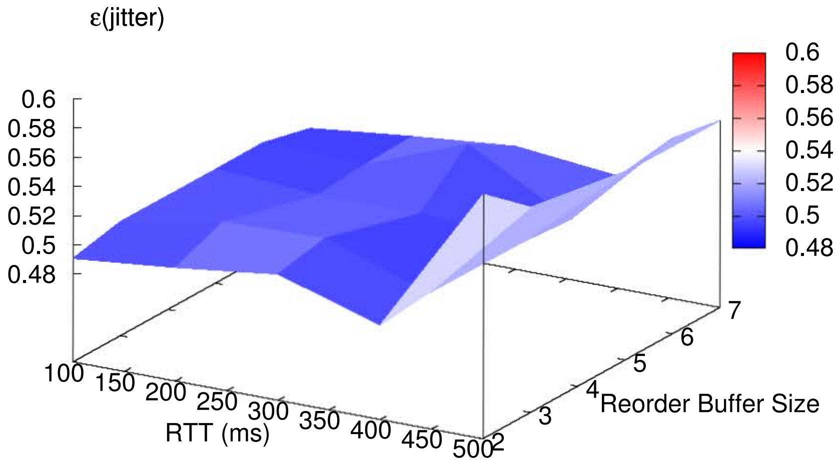


Figure 4.4: Jitter efficiency ratio

For voice applications, the E-Model's R factor quality metrics is computed based on the delay $I_d$ and equipment impairments $I_e$, the latter being related to the specific voice codec's quality and ability to handle losses:

$$R = 94.2 - I_d - I_e$$

with:

$$I_d = 0.024d + 0.11(d - 177.3)U(d - 177.3)$$

where U is a unity function: $U(x) = 0$ if $x < 0$ and 1 otherwise.

Video quality metrics $V_q$ is calculated using parameters which similarly relate to the video codec, however with a direct relationship defined for the frame and bit rates which can be varied, and to packet loss rate. We note that [160] only specifies QoE calculation for H.264 video codec being displayed on specific screen sizes.

$$V_q = 1 + I_{coding} exp\left(\frac{P_{plV}}{D_{PplV}}\right)$$

where $I_{coding}$ represents the basic video quality related to the coding distortion under a combination of video bit rate and frame rate, the packet loss robustness factor $D_{PplV}$ relates to codec robustness to packet loss and $P_{plV}$ represents the loss rate. Streaming video quality only degrades with delay greater than 100 msec [157].

MOS values in the range of 5–1 respectively represent Excellent, Good, Fair, Poor and Bad quality. [29] and [160] define the corresponding R factor and $V_q$ values.

We show example MOS values for VoIP and video applications using DCCP with and without the reordering buffer. We note that both voice and video applications may have a buffer to handle jitter and optionally reorder packets. To calculate MOS, we use simulation results for the DCCP rate available to the application and consider the additional delay introduced by either the DCCP or the application buffer. If out of order packets are not ordered by either of the buffers, they are considered lost by the application and included in MOS calculation.

Table 4.2 shows MOS for G.711 and G.729 voice codecs on a network with a 5% reorder rate, a 100 msec RTT and a buffer size of 5. It can be observed that buffering and reordering the data in either the transport protocol or the application has similar positive impact on the QoE of a voice conversation. We note that, due to low data rate of VoIP, the decreased DCCP rate is still high enough not to cause data losses.

| Options used | G.711 | G.729 |
|---|---|---|
| DCCP buffer | 4.33 | 3.96 |
| DCCP, no buffer; voice buffer | 4.33 | 3.96 |
| DCCP, no buffer; no voice buffer | 3.87 | 3.27 |

Table 4.2: MOS values for G.711 and G.729 voice codecs

Video applications have a higher data rate compared to VoIP and packet reordering resulting in a lower achievable DCCP rate may significantly impact the quality. Table 4.3 shows the MOS values for a H.264 codec video stream with a 5% reorder rate, 100 msec RTT and a buffer size of 12. We note the simulation results with the rate available to the video application being 898kbit/sec for our proposal and 314kbit/sec for standard DCCP. For a fair evaluation, we assume that the video application may have the capability to adjust the codec rate based on DCCP sender rate. Therefore, there are three cases: our proposal; standard DCCP with a rate adjustable video codec including an application buffer; and standard DCCP with a video buffer but no rate adjustment.

It can be observed that the proposed transport protocol buffer has the potential to greatly increase the applications QoE and that, in the least favourable scenario when a low rate application is used, it will produce quality similar to the standard DCCP protocol. When used with video, it will significantly improve an otherwise unacceptably low quality.

| Options used | MOS 4.2″ screen |
|---|---|
| DCCP-buffer | 4.19 |
| DCCP-no buffer; video buffer, rate adjustment | 2.76 |
| DCCP-no buffer; no video buffer, no rate adjustment | too low to calculate |

Table 4.3: MOS values for H.264 video codec

## 4.4 CONCLUSION

We have analyzed the performance degradation issues caused by packet reordering in unreliable transport protocols like DCCP and in multimedia applications. To combat these issues, we have proposed and evaluated the performance improvements achievable by a flexible receiver buffer implemented within a transport protocol. Our results show that this buffer allows to both increase the performance of the transport protocol and the QoE of the multimedia applications' user. We are currently considering to implement this scheme within the DCCP-TP [161] implementation.

# 5 EVALUATION OF CMT-SCTP IN ASYMMETRIC NETWORKS

Performance of multi-path transport protocols is known to be sensitive to path asymmetry. The difference between each path in terms of bandwidth, delay and packet loss has a potential to significantly decrease the overall performance of a data flow carried over multiple asymmetric paths. In this Chapter, we evaluate and analyse reliable data transfer in Concurrent Multipath Transfer extension of Stream Control Transport Protocol (CMT-SCTP) under various conditions of network asymmetry, with a focus on the use case where 3G and Wi-Fi networks are simultaneously available. We identify various causes of performance degradation, review the impact of CMT-SACK extension under path asymmetry and show that the total achievable goodput of a reliable in-order data flow over multiple heterogeneous paths is ruled by the characteristics of the worst path as perceived by the transport protocol. To support our study, we derive a simple analytical model of the receiver window blocking and validate it via simulation.

As multihomed mobile hosts downloading content from web based servers are our primary use scenario for multi-path transport, we consider the receiver buffer size to be comparatively smaller than the sender buffer and therefore the limiting factor. However, our analysis and conclusions about performance degradation are equally applicable to the case of mobile uploads where the reverse holds, i.e. the sender side buffer size is the limitation. In this chapter, we first illustrate the observed performance degradation and then pinpoint receiver window blocking, spurious retransmission, delayed SACK and incorrect RTT estimation as the causes of the degradation. Our contribution differs from Dreibholtz et al. who have only studied window blocking comprehensively and identified and provided a solution for bursts of spurious retransmissions caused by path change [162, 30]. Our key contributions in this chapter are the following.

- We propose a model of receiver's window blocking in CMT-SCTP which is related to round-robin packet scheduling and dissimilar congestion window growth on multiple asymmetric paths.
- We provide an analysis of a newly identified cause of spurious retransmissions, created by selective acknowledgement (SACK) packet reordering.

- We demonstrate the negative impact of the proposed SACK variant in CMT-SCTP [14] on the achievable performance and show that using the standard SCTP SACK mechanism can provide solid performance gains in cases of delay and bandwidth asymmetry.

- We identify incorrect return trip time (RTT) estimation as an additional cause of performance degradation.

The organization of this chapter is as follows. In Section 5.1 we provide and overview of the CMT-SCTP protocol, our simulation environment and results of our experiments with a range of network asymmetry parameters. In Section 5.2 we analyse causes of performance degradation in CMT-SCTP on asymmetrical paths. We draw conclusions and discuss future work in Section 5.3.

## 5.1 CMT-SCTP IN ASYMMETRIC MULTIPATH ENVIRONMENT

SCTP was proposed to overcome the design limitations of TCP and UDP with support for multi-homing, multi-streaming, message oriented data delivery and congestion controlled fully or partially reliable data delivery [23]. The CMT extension of SCTP [14] enables the simultaneous use of multiple physical network paths for the same logical data connection.

Packets are distributed between the paths using a round-robin mechanism and therefore each path is presented with an equal opportunity to transmit data, within the limits imposed by the congestion control mechanism.

CMT-SCTP inherits SCTP mechanisms. Congestion control is performed with a TCP-like window based algorithm, with the sender congestion control window (CWND) maintained separately for each path. Data receiver maintains a single receiving window (RWND) for all paths. At any time instant $t_i$ when there is an update of the CWND size, resulting from any of the mechanism components, and for each path $k \in \mathbb{M}$, the resulting value $CWND_{i,k}$ needs to satisfy the following equation.

$$CWND_{i,k} = min(CWND_{i-1,k}, RWND_{i-1}) \tag{5.1}$$

The number of packets which are stored in the receiver's buffer never exceeds the maximum receiver's window i.e. the receiver's buffer size.

CMT-SCTP includes partially or fully reliable transport with in-order or out of order packet delivery. For reliable transport with in-order delivery which is the primary focus of this chapter, successful reception of data packets is acknowledged by the receiver using selective acknowledgement (SACK) packets. SACK packets include the information about the received

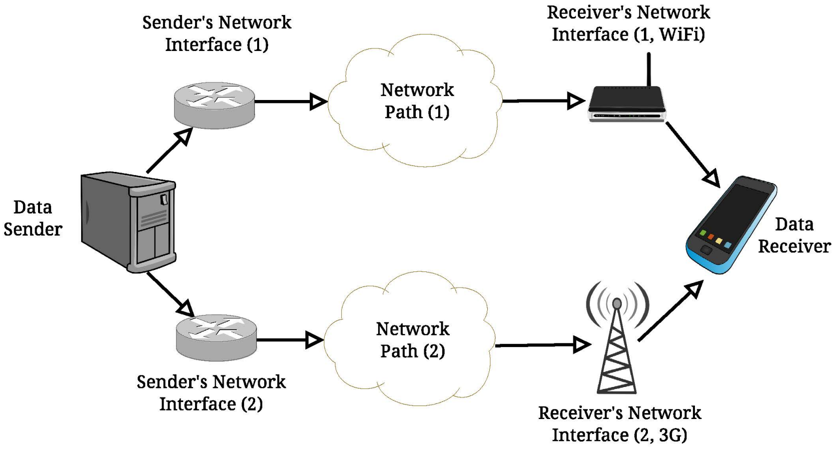

Sender's Network Interface (1)
Network Path (1)
Receiver's Network Interface (1, WiFi)
Data Sender
Data Receiver
Network Path (2)
Sender's Network Interface (2)
Receiver's Network Interface (2, 3G)

Figure 5.1: Simulated Network Topology

in-order packets and gap reports, representing the received and yet be released out of order packets queued in the receiver's buffer. Unlike TCP, packets received over any available path within a single data connection can be acknowledged by the receiver over any available path.

SCTP, similarly to TCP, includes two modes of packets retransmission. Fast retransmission is triggered if a packet sequence (TSN) is reported as missing in three SACK gap reports. SCTP then enters Fast Recovery (FR) mode where the congestion window (CWND) of the corresponding path is adjusted to the half of the current value. A timeout based retransmission is triggered when no SACK has been received within a retransmission timeout (RTO) period. The RTO value is computed from the smoothed RTT (SRTT) for the particular path. The penalty for RTO expiry is greater compared to fast retransmission as the timeout forces SCTP to go into slow-start phase for the corresponding path.

CMT-SCTP [14] proposes a modification of the standard SCTP SACK mechanism, to minimise control traffic (the number of SACK messages) on the return channel. For the sake of clarity, we will refer to the CMT modifications [14] as CMT-SACK. As these effectively reduce the timeliness of received SACK information, in the remainder of the chapter we investigate the performance of CMT-SCTP with both SACK options in the environment with asymmetric network paths.

### 5.1.1 *Simulation Environment*

To evaluate the performance of reliable transmission of a single CMT-SCTP flow on multiple asymmetrical paths, we use the network simulator NS-2 [113] version 2.35 with the most recent CMT-SCTP patch set.

For all simulations, we use a simple network topology shown in Figure 5.1 with one data sender and one receiver, each equipped with two network

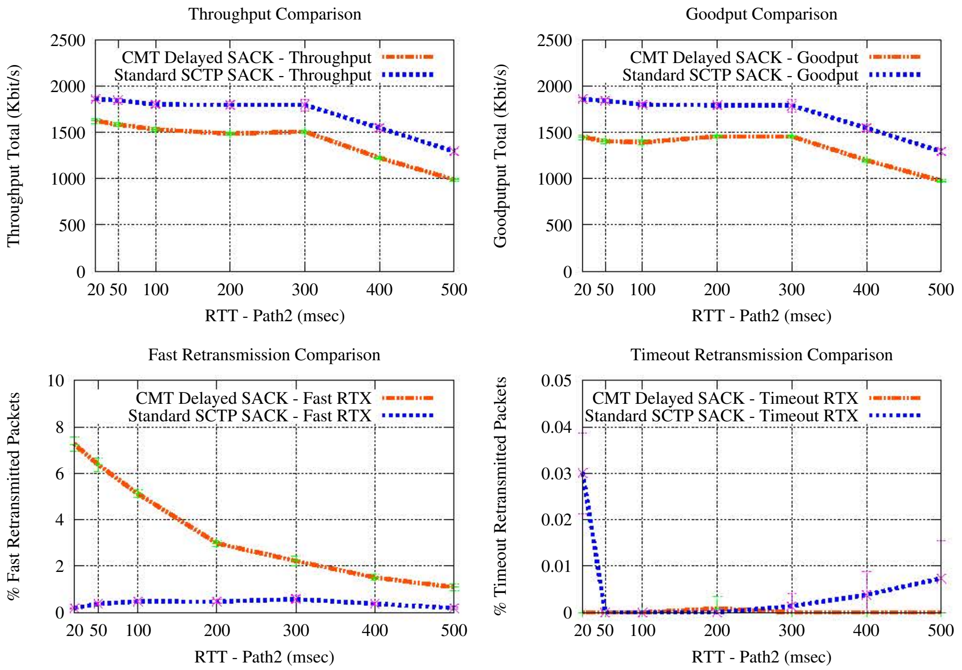


Figure 5.2: Impact of RTT asymmetry on the performance of CMT-SCTP with standard SACK and CMT-SACK options

interfaces. We vary RTT, packet loss and bandwidth in specific simulations to introduce path asymmetry. We additionally evaluate the effects of path asymmetry for the common case where 3G and Wi-Fi networks are concurrently used. We note that, although our experiments concentrate on a two path case, the results can be generalised to any number of asymmetrical concurrent paths.

In all simulations, we have used the default receiver's buffer size of 65536 bytes. We note that the bandwidth delay product (BDP) for all simulation parameter combinations is always smaller than the receiver buffer size, therefore this is not a limiting factor in the performance of the transport protocol.

In the following, we present the performance of reliable transfer CM-SCTP with the standard SCTP and CMT-SCTP SACK mechanisms. All presented results are averaged over 10 simulation runs, each with a duration of 3 minutes and with different random seed values.

### 5.1.2 *RTT Asymmetry*

In Figure 5.2 we present simulation results for RTT asymmetry between network paths. Both paths have a bandwidth of $1\mathtt{Mbit/s}$, RTT for $\mathtt{Path}_1$ is fixed to $20\mathtt{msec}$ and we vary RTT on $\mathtt{Path}_2$ between $20\mathtt{msec}$ and $500\mathtt{msec}$. We present the resulting aggregate throughput and application goodput for

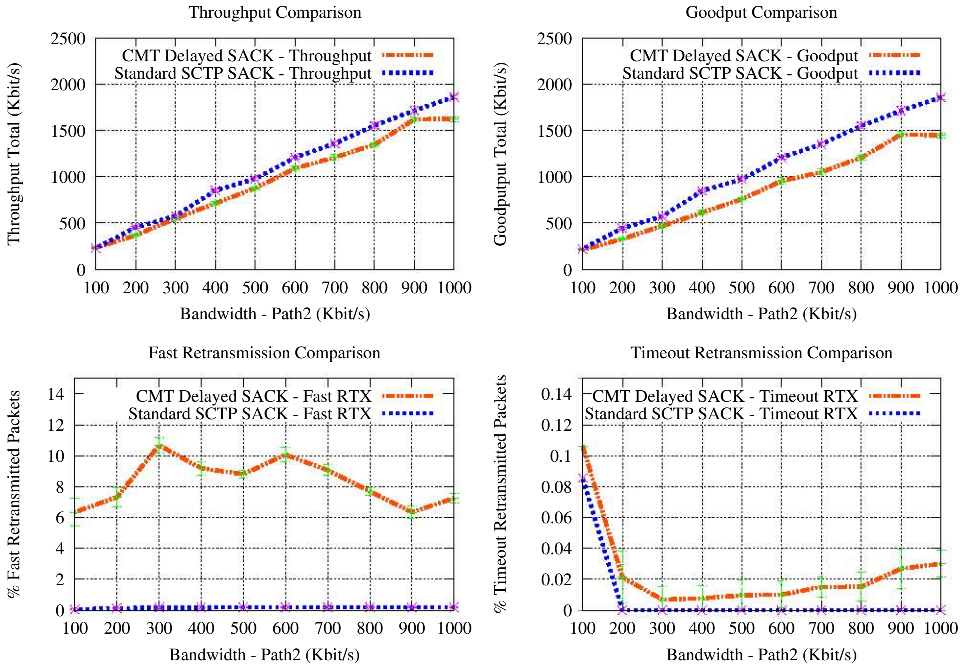


Figure 5.3: Impact of bandwidth asymmetry on the performance of CMT-SCTP with standard SACK and CMT-SACK options

both CMT-SACK and non-delayed SACK. The throughput and goodput are measured, respectively, at the receiver input and at the output of the receiver buffer. We also show the corresponding number of packet retransmissions. It can be observed that as the RTT asymmetry between the paths grows, the total throughput and goodput gradually degrade from close to the total available capacity of both paths ($2\mathtt{Mbit/s}$) to the value equivalent to what is available on a single path ($1\mathtt{Mbit/s}$). The SACK option has, on the average, a 27.8% higher goodput (and similar for throughput) to that was measured using CMT-SACK. We can observe that using the standard SACK option reduces the number of retransmitted packets to an amount close to zero (in the lower graphs in Figure 5.2).

### 5.1.3 *Bandwidth Asymmetry*

In Figure 5.3, we present simulation results for the bandwidth asymmetric network scenario. In these simulation runs, RTT of $20msec$ was assigned to both paths. $\mathtt{Path}_1$ has a capacity of $1\mathtt{Mbit/s}$ while the capacity of $\mathtt{Path}_2$ was varied from $1\mathtt{Mbit/s}$ to $100\mathtt{Kbit/s}$. As can be observed, as the bandwidth asymmetry grows, the overall throughput and goodput again gradually degrade towards the values which would be achievable on the lowest capacity path. Again there is a notable performance improvement for the configuration using the standard SACK (25.8% on the average) compared

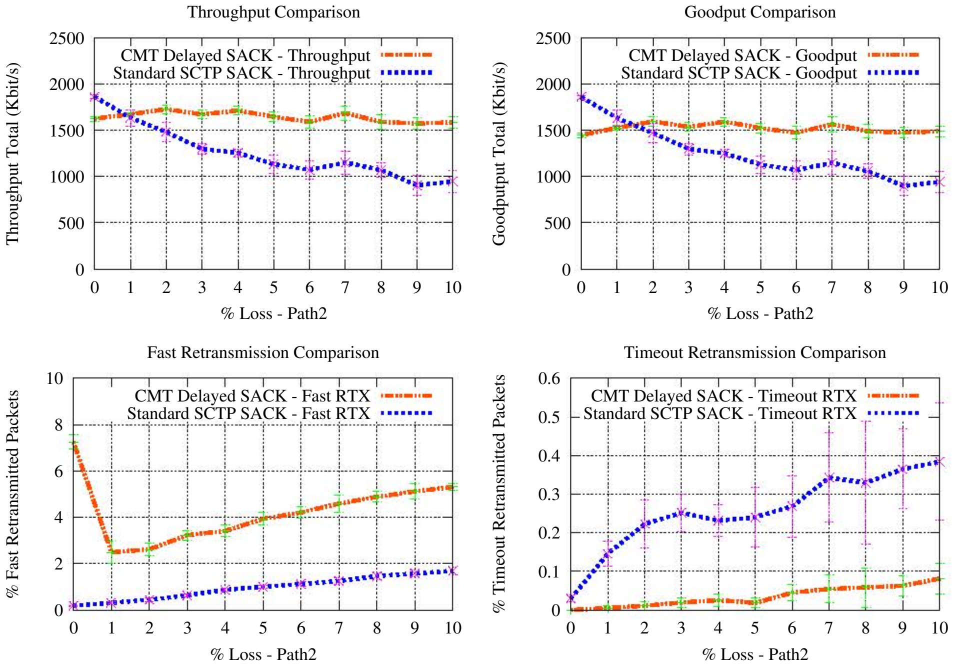


Figure 5.4: Impact of loss asymmetry on the performance of CMT-SCTP with standard SACK and CMT-SACK options

to the results obtained by using CMT-SACK. We also note the higher number of retransmissions (again largely fast retransmissions) than in the case of RTT imbalance.

### 5.1.4 *Loss Asymmetry*

Figure 5.4 presents simulation results for loss asymmetry between network paths. In this scenario, bandwidth and RTT for both paths are, respectively, 1Mbit/s and 20ms. $\mathtt{Path}_1$ has no loss while $\mathtt{Path}_2$ has a packet loss rate (PLR) varied between 0% and 10%. As can be observed in Figure 5.4, in the case of CMT-SACK, the impact of packet loss on the performance degradation is minimal. We speculate that in some cases packet loss actually may help to minimise the effects of receiver's window blocking by triggering faster retransmission of the missing packets in the receiver's buffer.

In the lower graph of Figure 5.4, we present a detailed breakdown of retransmitted packet types, showing the number of fast retransmission and time-out retransmissions. As can be seen, for all cases of packet loss on $\mathtt{Path}_2$, CMT-SACK triggers a higher number of fast retransmissions than standard SACK. In the case of loss imbalance, this is actually advantageous as it minimises the likelihood of the retransmission timeout (RTO) triggers which results in a bigger penalty in CWND decrease. We note that in all cases of packet loss on $\mathtt{Path}_2$, there are occasional time-out retransmission events and the number of these events is higher for the SACK mechanism.

In conclusion, although the SACK mechanism provides significant performance gains over CMT-SACK for RTT and bandwidth path asymmetry, it does not handle any but the very small loss rate imbalance. It also increases the return traffic volume i.e. the maximum observed value of the ratio of the return traffic volume and goodput, comparing SACK and CMT-SACK simulation results, was 5%. Therefore, the SACK mechanism variant which would provide a good performance for all cases of path imbalance is a challenge for future study.

## 5.2 DEEP DIVE INTO PERFORMANCE DEGRADATION

In CMT-SCTP with any kind of path asymmetry, the growth of congestion windows, driven by the path parameters on the different paths will also be asymmetric. The first cause of windows growth limitation and corresponding reduction in data rate follows from equation (1). This will limit the size of sender window on any of the paths to the smallest window size across all paths. Second, the uneven data release on specific paths will cause out of sequence or delayed arrival of packets from slower (or longer delay) paths. This will progressively reduce the available receiver window size and according to equation (1), the sender window sizes on all paths, until finally the receiver window is blocked and all data transfer stalls.

### 5.2.1 *Receiver's Window Blocking*

In this section, we will first calculate the lower bound of the receiver window blocking time duration using the proposed analytical model. This represents the case where the receiver window has recovered naturally, i.e. the missing packets have arrived without any retransmissions. The next step up in severity of problems resulting from imbalance is the triggering of fast recovery, where 3 SACK packets were received with gap reports for the same TSN. This reduces the sender congestion window and triggers the retransmission of the missing TSN. The worst case for the reduction of congestion window size is the expiry of RTO, which additionally triggers a slow start phase.

#### 5.2.1.1 *Minimum Receiver's Window Blocking Time Model*

We present an analytical model which represents the lower bound of the receiver's window blocking time. Higher values for each packet $P_n$ with $n \in \mathbb{N}$, we denote the packet departing time as $a_n$ and the transmission delay of packet, regardless of the path used, as $D_n$. We note that the path delay characteristic is embedded in $D_n$.

Considering that each packet has a fixed size $L$ and assuming $C_k$ is the effective throughput of path $k$ with $k \in \mathbb{N}^*$, we have:

$$a_n = \frac{n.L}{C_k} \tag{5.2}$$

Now considering a given receiver's window, the minimum blocking time of a packet $P_i$ depends on all sent, but yet to be received (in flight) packets $(P_j, P_{j-1}, P_{j-2}, ...)$ as illustrated in Fig. 5.5.

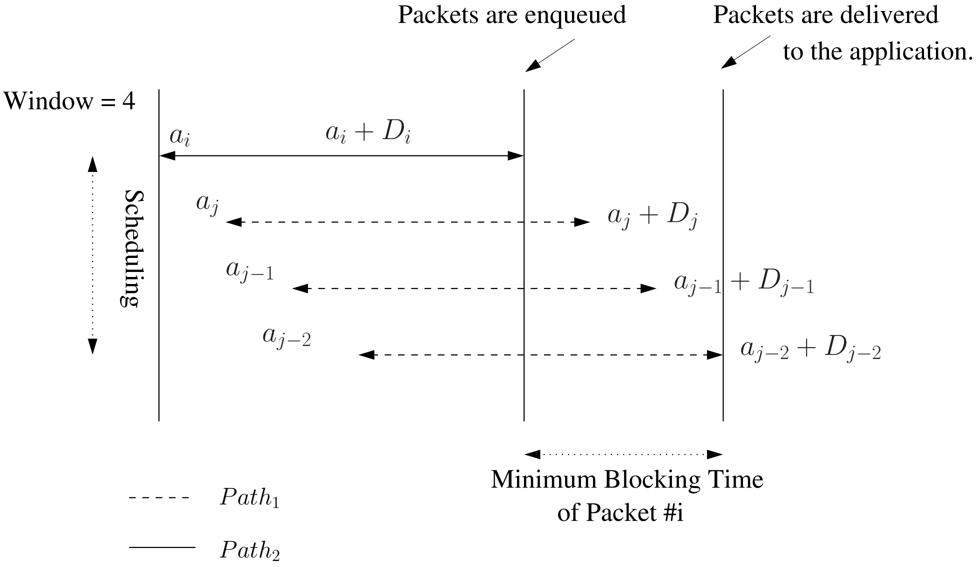


Figure 5.5: Blocking model illustration

As a result, the blocking time $T_{block}$ of packet $P_i$ is given as follows:

$$T_{block} = \max_i \{0, \max_{j<i}(a_j + D_j) - (a_i + D_i)\} \tag{5.3}$$

Taking as an example the sequence number progression of a connection, each $T_{block}$ should correspond to a stall period. The receiver window will stay blocked until either the delayed packets arrive, as per equation (2), retransmission is triggered by fast recovery, or by reaching the RTO duration and triggering the timeout based retransmission.

#### 5.2.1.2 *Validating the Model*

We compare analytical results with values for receiver window blocking duration derived by simulation, for a range of parameter values. First, we simulate the equal bandwidth, RTT-asymmetric scenario where both paths have $1 \mathrm{Mbit/s}$ capacity, $\mathrm{Path}_1$ has an RTT of $20 \mathrm{msec}$ and $\mathrm{Path}_2$ has an RTT between $20 - 500 \mathrm{msec}$. The second scenario is of equal delay, bandwidth

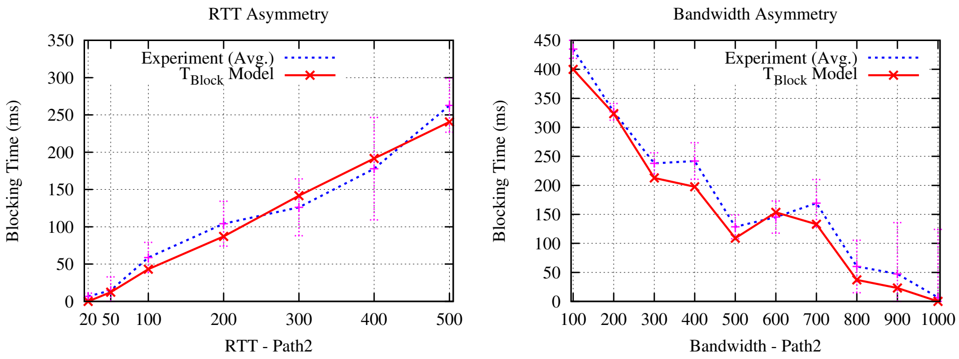


Figure 5.6: Comparison of analytical model and simulation results for the receiver window blocking time duration; RTT asymmetry and bandwidth asymmetry scenarios

| $T_{Block}$ | Model | Experiment Avg. | Experiment Std.Dev. |
|---|---|---|---|
| (ms) | 62.82 | 72.34 | 9.52 |

Table 5.1: A comparison of analytical and simulated $T_{Block}$ values for the 3G - Wi-Fi configuration

asymmetry, where both paths have and RTT of $20msec$, $Path_1$ has a capacity of $1Mbit/s$ and the bandwidth on $Path_2$ is varied. Both in simulations and in the analytical model we use a packet size of $1460Bytes$. Simulation results were derived by averaging ten $1min$ duration runs, with different seed values. The results shown in Figure 5.6 show a very good match between the analytical and simulation results.

We additionally calculate the receiver window blocking time duration values for a CMT-SCTP scenario when concurrent paths are on 3G and Wi-Fi networks, having both bandwidth and RTT asymmetry. We assume access to a local server and a a Wi-Fi hotspot scenario, where the available bandwidth is limited to $1Mbit/s$ and the (correspondingly low) $RTT = 20ms$. 3G parameters are based on the results reported in [98], with an available bandwidth of $400kbit/s$ and an $RTT = 100msec$, also assuming local server access. We note that our own experiments on 3G in Sydney, Australia, have indicated a higher delay, with RTT values of around $150msec$ and similar available bandwidth, thus confirming the asymmetry between the two networks. Table 5.1 shows the values obtained from the proposed analytical model and using simulation, which are again very close.

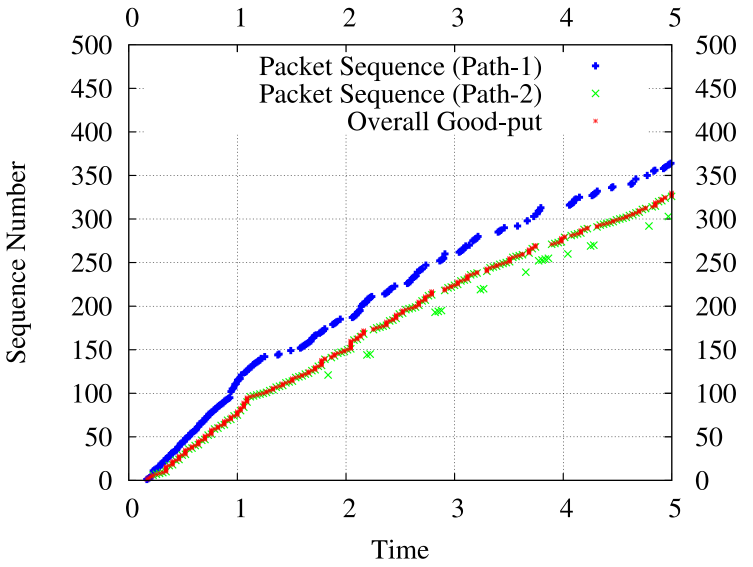


Figure 5.7: Packet Sequence Progression in Asymmetric CMT-SCTP Transmission

### 5.2.2 *Path Asymmetry and Goodput Variations*

We note that, as a consequence of receiver window blocking and other issues created by the concurrent path imbalance (to be detailed later in this section), a significant amount of buffering takes place in the receiver. In addition to the reduction in the overall throughput and goodput, as compared to what may be achieved on each path separately, shown in Figures 5.2, 5.3 and 5.4, the buffering and releasing of packets introduces artificial jitter and delay for the upper layer protocols and has a potential to further impact the overall performance CMT-SCTP under asymmetric network conditions.

In Figure 5.7 we present the first 10 seconds of a simulation showing how the overall goodput is ruled by the low performing path in an asymmetric CMT-SCTP transmission scenario. We again mimic a multi-path scenario with a mobile having Wi-Fi and 3G connectivity, using the same parameters as in Figure 5.9. To mimic a lossy Wi-Fi channel we additionally introduce 1% uniformly distributed packet loss.As can be observed in Figure 5.7, the over-all goodput data, which are the packets available to the application layer, is clearly controlled by 3G ($\mathtt{Path_2}$), the low performing path (in regards to bandwidth and delay) and there is a perceivable jitter in the goodput data. In addition to the underutilisation of the available resources, the jitter has a potential to further degrade the performance of real time application like video streaming.

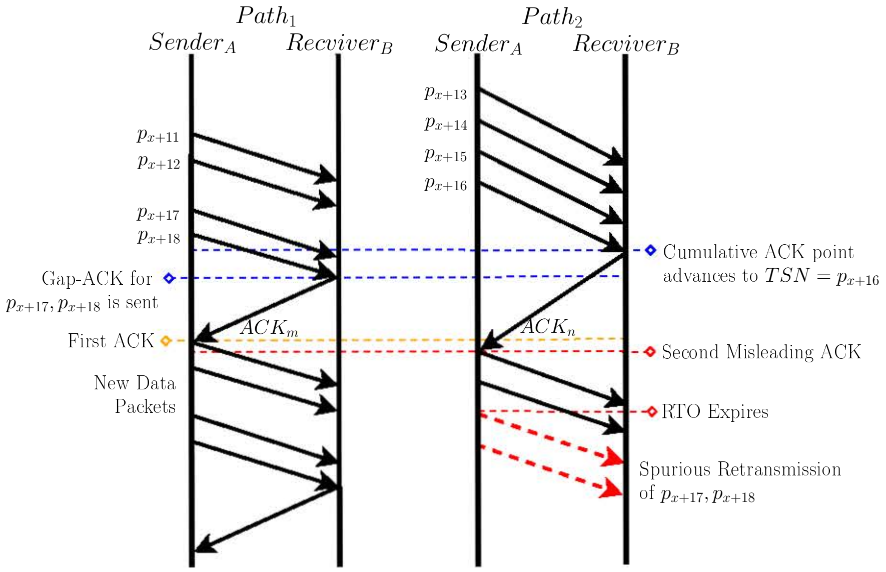


Figure 5.8: Unordered Arrival of SACK Triggering Spurious Retransmission

### 5.2.3 *Spurious Retransmission*

In [162] the authors have identified a performance degradation due to bursts of spurious retransmissions caused by path changes. Here, we highlight a second cause of spurious retransmissions, resulting from the acknowledgement (SACK) packets arriving at the sender out of order due to asymmetry in return path characteristics.

In Figure 5.8, we present an illustration of how spurious retransmission is triggered on multiple packets due to out of order received SACK packets in a multi-path data transmission session. We consider that sender (A) is transmitting data towards the receiver (B) over two paths $\mathrm{Path}_1$ and $\mathrm{Path}_2$. The paths differ both in terms of bandwidth and RTT. For simplicity of the illustration, we ignore packet loss.

As can be seen in Figure 5.8, acknowledgement packets $\mathrm{SACK}_m$ over $\mathrm{Path}_1$ and $\mathrm{SACK}_n$ over $\mathrm{Path}_2$ are both transmitted towards the data sender with a small time delay. $\mathrm{SACK}_m$ contains information about cumulative acknowledgement of packets upto 16. $\mathrm{SACK}_n$ contains information of cumulative acknowledgement up to 12 and gap reports for packets $13-16$. When these SACK packets are received out of order at the sender, as a result the sender will have incorrect information about packets in the receiver's buffer and about the size of the receiver's window. SCTP-CMT sender makes decisions based on the SACK packet which has the highest cumulative acknowledgement [23]. As CMT-SCTP allows ACK packets to be received over any available path, the incorrectly reported out of order SACK would trigger spurious retransmission of already accumulated packets in the receiver's buffer, based on two triggers: (a) the received SACK packet, which is responsible for triggering of fast-retransmission and (b) per path RTO expiry, which triggers timeout retransmission.

We note that Non-Renegable SACK (NR-SACK) [163], [164], proposed by Nataranjan et al. targets performance improvements. However, the authors [164] do not evaluate the mechanism under an asymmetric RTT and bandwidth scenarios similar to ours. Furthermore, in our NS-2 experiments with NR-SACK enabled CMT, we have observed less than expected gain with NR-SACK under asymmetric scenarios and throughput was still found to be limited by the worst perceived path. Finally, TCP uses the F-RTO mechanism [165] as part of the congestion control algorithm to detect spurious time-out and packet retransmissions, however it is also prone to packet reordering related performance degradation.

### 5.2.4 *Incorrect RTT Estimation*

RTT and RTO calculation in SCTP is performed in the same way as in TCP using Smoothed RTT ($SRTT$) and RTT Variance ($RTTVAR$) variables [23] as shown in equation (5.4). Each time a SACK packet is received, these variables are updated using the most recent RTT estimation $R'$. $RTO_\alpha$ and $RTO_\beta$ are set to $\frac{1}{8}$ and $\frac{1}{4}$ as recommended in [166]. In case of CMT-SCTP, the sender continuously estimates these variables for each destination address. Therefore, in a CMT-SCTP data flow which uses two paths, estimation of $SRTT$, $RTO$ and $RTTVAR$ variables will be carried out for both paths.

$$\begin{aligned} RTO &= SRTT + 4.RTTVAR. \\ SRTT &= (1 - RTO_\alpha).SRTT_{last} + RTO_\alpha \times R' \\ RTTVAR &= (1 - RTO_\beta.RTTVAR_{last}) \\ &\quad + RTO_\beta \times |SRR - R'| \end{aligned} \tag{5.4}$$

In Figure 5.9, we present results from an asymmetric RTT CMT-SCTP experiment on concurrent 3G and Wi-Fi paths, with both CMT-SACK and SACK configurations, with simulation parameters as per the 3G–Wi-Fi scenario used to validate receiver window blocking in the previous subsection. As can be observed, in case of both CMT-SACK and standard SACK, RTT for Wi-Fi ($Path_1$ ) is always estimated incorrectly, around $128ms$ for the CMT-SACK and $132ms$ for SACK (note the true network RTT for Wi-Fi is $20ms$). Incorrect estimation of RTT will clearly lead to inaccurate estimation of RTO as shown by the equation 5.4 in case of asymmetric CMT-SCTP transmission. This incorrect estimation may be partially prevented by the SCTP RFC's [23] recommendation of setting a minimum RTO to $1sec$. But this may be counter-productive for the performance of certain paths of a CMT-SCTP connection in case of lossy forward and reverse channel, where

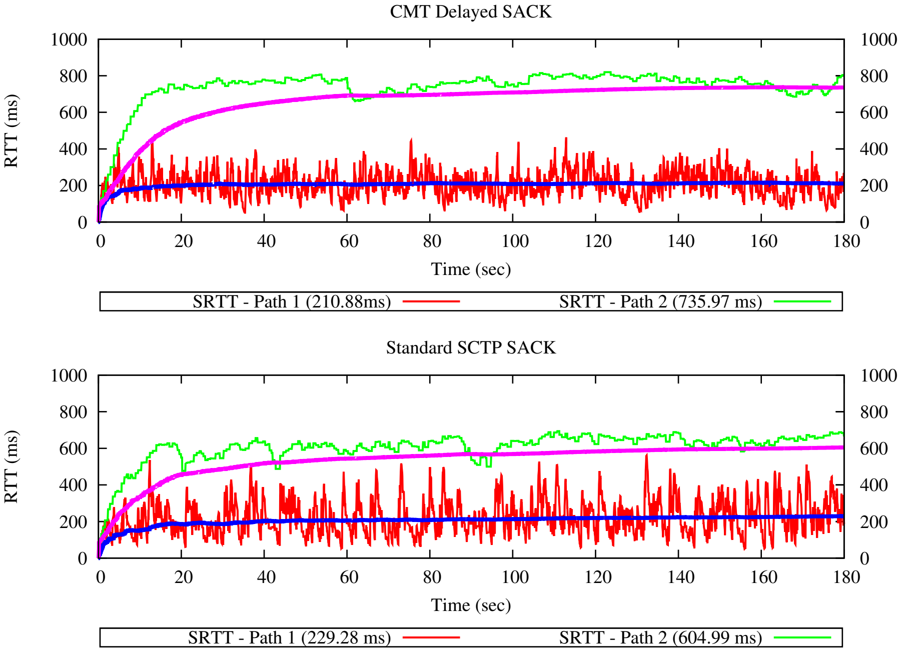


Figure 5.9: Estimated Smoothed RTT in an Asymmetric CMT-SCTP Data Transmission

the SACK packets will fail to trigger the faster-retransmission algorithm and unnecessarily delay the retransmission of the lost packet.

## 5.3 CONCLUSION

In this chapter, we have presented the evaluation of the CMT-SCTP performance on dissimilar network paths and provided a detailed analysis of the causes of performance degradation including a model of receiver's window blocking. Although our simulations concentrate on a two path case, the results are equally applicable to any number of asymmetrical concurrent paths. In future work, we plan to investigate mechanisms which would mitigate the effects of network path asymmetry, with the aim of enabling efficient concurrent use of heterogeneous networks.

# 6 DELAY AWARE SCHEDULING TO MITIGATE RWND BLOCKING

Reliable in-order multi-path data transfer under asymmetric heterogeneous network conditions has known problems related to receiver's buffer blocking, caused by out of order packet arrival. Consequently, the aggregate capacity from multiple paths, which theoretically should be available to and achievable by the multi-path transport protocol, is practically severely underutilized. Several mitigation techniques have been proposed to address this issue mostly by using various packet retransmission schemes, load-balancing and bandwidth-estimation based mechanisms. In comparison to the existing reactive techniques for buffer block mitigation, we propose a novel and yet simpler to implement, delay aware packet scheduling scheme for multi-path data transfer over asymmetric network paths, that proactively minimizes the blocking inside receiver's buffer. Our initial simulation results show that, in comparison to the default round robin packet scheduler, by using our proposed delay aware packet scheduling scheme, we can significantly improve the overall performance of a multi-path transport protocols while notably minimizing the receiver's buffer usage. Therefore, our proposal is particularly beneficial for multi-homed hand-held mobile devices with limited buffering capacity, which, due to their multi-homing and heterogeneous wireless network features (i.e. availability of 3G and Wi-Fi) are also one of the most common use cases for multi-path transport.

The organization of the rest of the chapter is as follows: in Section 6.1 we propose and explain our delay-aware packet scheduling in contrast to CMT-SCTP's default round-robin packet scheduling. We additionally present a proposal for a delay estimation mechanism and the required modifications to CMT-SCTP. In Section 6.2 we present an evaluation of the proposed packet scheduling technique and compare it with CMT-SCTP's sender congestion window round robin packet scheduler. We also present the impact of a varying delay estimation error to the performance of our proposal. Finally in Section 6.3 we draw conclusion and propose future work.

## 6.1 DELAY AWARE PACKET SCHEDULING FOR MULTI-PATH DATA TRANSFER

In this section, we first introduce the round robin scheduling currently utilized in the CMT-SCTP transport protocol and the associated performance issues. We then present our proposal for delay based packet scheduling, followed by the description of a mechanism used to estimate the delay on specific paths.

### 6.1.1 *Round Robin Scheduling*

In CMT-SCTP's original round robin packet scheduling mechanism, data sender attempts to send packets on multiple paths, based on the congestion window of each path. A separate congestion window is kept for each individual path. The scheduler observes the usable partition of sender's congestion window, subtracting the on-the-flight unacknowledged packets from current congestion window size for each path, in a so called "blind" round robin manner using the Equation 6.1 where $Cwnd_i$ and $Unacknowledged_i$ represents the sender side congestion window size and amount of unacknowledged packets for any path $P_i$. The resulting amount of packets from Equation 6.1, which is assumed to be safe to prevent the overflow of the receiver's buffer, is transmitted over the particular path during multi-path transmission. Each path gets the opportunity to transmit packets based on the output of Equation 6.1 in a round robin manner. Once the receiver's buffer is full with out of order packets i.e. with $Rwnd = 0$, the missing packets are retransmitted, even though they might actually be in flight over the longer delay paths. The retransmission based mitigation techniques perform their role in these cases by determining the best path to retransmit the missing packet(s) based on various path characteristics criteria which will lead to unblocking of the receiver's buffer [100, 101, 102].

$$min(Cwnd_i - Unacknowledged_i, Rwnd) \tag{6.1}$$

An illustration of buffer block in round robin packet scheduling in presented in Table 6.1 where the multi-path transport protocol is using two paths: $P_1$ and $P_2$, with $RTT_1 = 20ms$ and $RTT_2 = 200ms$ respectively. The bandwidth of the paths are considered $B_1 = 1.6Mbit/s$ and $B_2 = 400Kbit/s$ respectively. The above parameters closely represent a multi-path scenario, where a mobile device e.g. smartphone which has two (heterogeneous) interfaces, 3G and Wi-Fi and is downloading content from a data-centre.

Therefore, considering symmetric forward and reverse delay on both paths, with an average packet size of $S = 1000Bytes$, $P_1$ will emit $\frac{B_i \times RTT_i}{8 \times S} = \frac{1.6 \times 1000 \times 1000}{8 \times 1000} \times \frac{10}{1000} = 2Packets$ at every $10ms$. Similarly, $P_2$ will emit

| Xmit Time $(ms)$ | Xmit over $P_1$ | Xmit over $P_2$ | Rcvd. Time $(ms)$ | Rcvd. over $P_1$ | Rcvd. over $P_2$ | Good-put | Pkts in Rbuf | Remark |
|---|---|---|---|---|---|---|---|---|
| $0$ | $1-2$ | $3-7$ | $10$ | $1-2$ | $none$ | $1-2$ | $0$ | |
| $10$ | $8-9$ | $none$ | $20$ | $8-9$ | $none$ | $none$ | $2$ | |
| $20$ | $10-11$ | $none$ | $30$ | $10-11$ | $none$ | $none$ | $4$ | |
| $30$ | $12-13$ | $none$ | $40$ | $12-13$ | $none$ | $none$ | $6$ | |
| $40$ | $14-15$ | $none$ | $50$ | $14-15$ | $none$ | $none$ | $8$ | |
| $50$ | $16-17$ | $none$ | $60$ | $16-17$ | $none$ | $none$ | $10$ | Blocked |
| $\dots$ | $\dots$ | $\dots$ | $\dots$ | $\dots$ | $\dots$ | $none$ | $10$ | Blocked |
| $90$ | $none$ | $none$ | $100$ | $none$ | $3-7$ | $3-17$ | | Un-blocked |

Table 6.1: Traditional Round-Robin Packet Scheduling

$\frac{400\times1000}{8\times1000} \times \frac{100}{1000} = 5Packets$ at every $100ms$ until the receiver's buffer is full. For the sake of illustration, we assume that the receiver's buffer size is $10packets$ in Table 6.1.

As can be seen in Table 6.1, receiver buffer is blocked during $50ms - 100ms$ clearly due to the way packet sequences are selected by the scheduler. To further establish our argument on the problem, we present the NS-2 [113] simulation results for CMT-SCTP with round-robin scheduling in Figures 6.1 and 6.2. The parameters chosen for this simulation were the same as in Table 6.1.

Figure 6.1 presents the packet sequence progression in time[1] As can be seen, although we receive packets over both $Path_1$ and $Path_2$, the application-received packets are controlled by the worse bandwidth-delay path, due to the buffer-and-release nature introduced as a consequence of round-robin scheduling. Figure 6.2 depicts the throughput evolution in time (throughput is averaged over 1 second intervals), for each path and aggregated, i.e. presented to the application. Similarly, the reduced performance of the multi-path transport protocol can evidently be observed from the figure, as the transport protocol utilizes only a part of the available total capacity.

We should note that it is possible to get around buffer blocking by providing a large enough buffer. For the given scenario in Table 6.1, we will at least need a buffer with size equal to the combined bandwidth delay prod-

[1] Note that we only show the first 5 seconds of the data transfer.

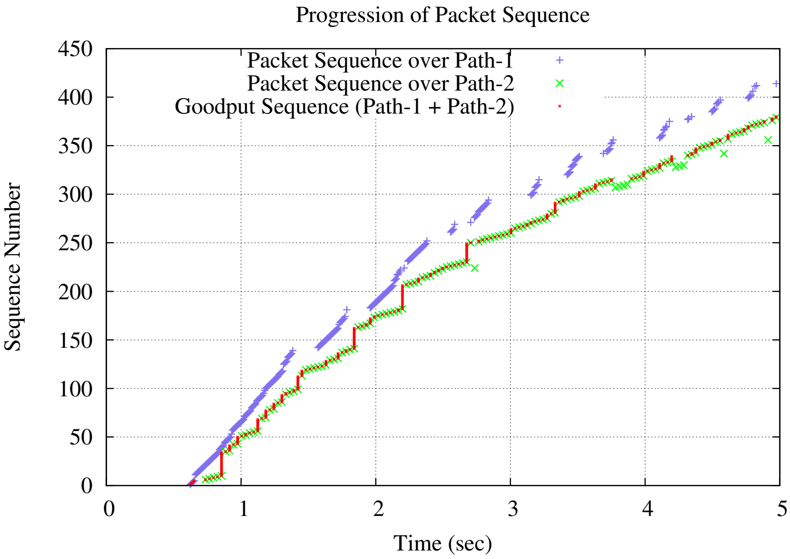


Figure 6.1: Impact of receiver's buffer blocking on application received packets

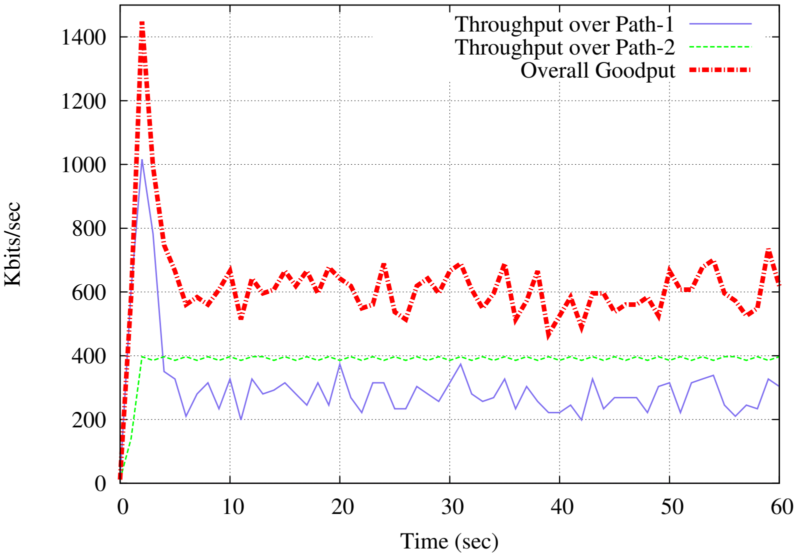


Figure 6.2: Impact of receiver's buffer blocking on the overall aggregated goodput

uct with respect to the highest RTT of the corresponding paths, as shown in Equation 6.2, to avoid blocking.

$$Rbuf_{min} = \sum_{i \in \{P_1, P_2, \ldots P_n\}} B_i \times \max_{j \in \{P_1, P_2, \ldots P_n\}} (RTT_j) \tag{6.2}$$

From Equation 6.2, a given receiver's buffer size of 50KB would have sufficed for the illustrated scenario in Table 6.1. But this would not be a scalable and optimum solution as $Rbuf_{min}$ easily becomes more demanding if we consider a slightly different scenario of two paths with 10Mbit/s and 1Mbit/s bandwidth and 20ms and 200ms RTT respectively, where the required minimum buffer size would be 275KB to avoid blocking. The required minimum buffer size will be even higher if we consider lossy scenarios where lost or delayed packets will frequently need to be retransmitted. Therefore, to avoid these existing problems with round-robin scheduling, we propose delay-aware scheduling in Section 6.1.2.

### 6.1.2 *Delay Aware Scheduling*

To mitigate the problems associated with the path unaware round-robin packet scheduling, in this section we propose a delay aware packet scheduling which carefully selects packet sequences to be transmitted over each path. The main idea behind delay aware packet scheduling is not to transmit monotonically increasing packet sequences in a multi-path transfer, but to carefully choose and then emit packets based on the delay of the associated paths to receive packets in order.

First, we define $P_i \in \{P_1, P_2, \ldots P_n\}$ as the set of the paths associated in a multi-path transmission and $D_i \in \{D_1, D_2, \ldots D_n\}$ are the respective forward delays of the paths. We assume that the set of the paths is sorted in ascending order based on their forward delays. The packet emission capacity of each $P_i$ is given by $C_i \in \{C_1, C_2, \ldots C_n\}$ which could be estimated from instantaneous congestion window of each path.

Then we obtain the ideal number of packets $K_i$ that can be transmitted on the path $P_i$ within $lcm$[2]$(D_i \in \{D_1, D_2, \ldots D_n\})$ time using Equation 6.3. In an ideal scenario, the time duration $lcm(D_i \in \{D_1, D_2, \ldots D_n\})$ ensures that having started at time instant 0, the scheduler will be back to the same state after $lcm(D_i \in \{D_1, D_2, \ldots D_n\})$ amount of time.

$$K_i = lcm\left(D_j \in \{D_1, D_2, \ldots, D_n\}\right) \times \frac{C_i}{D_i} \tag{6.3}$$

[2] Lowest Common Multiple (LCM)

Thus the ideal number of packets N sent on all the paths during the time $lcm\left(D_j \in \{D_1, D_2, ..., D_n\}\right)$ is given by Equation 6.4.

$$N = \sum_{i \in \{1,2,...,n\}} lcm\left(D_j \in \{D_1, D_2, ..., D_n\}\right) \times \frac{C_i}{D_i} \tag{6.4}$$

Our goal now is to transmit this N packets in such an order over the available paths that they would occupy the least amount of space in the receiver's buffer. To infer in order packet reception, we create the vector $O_i \in \{O_1, O_2, ..., O_{\sum_{i\in\{1,2,...,n\}} \frac{lcm(D_i)}{D_i}}\}$ that contains the ideal order of the paths in which the transmitted packets should be received. Calculation of $O_i$ is shown in Algorithm 1.

In order to prove our hypothesis, we considering a deterministic scenario where the delays of the paths do not change during the transmission of these N packets.

**Algorithm 1** Expected Order of Data Reception in Delay Aware Scheduling

```
$j \leftarrow 0$
$t \leftarrow 1$
while $t \leqslant lcm(D_i \in \{D_1, D_2, ..., D_n\})$ do
    for each $P_i \in \{P_1, P_2, ..., P_n\}$ do
        if $t \equiv 0 \pmod{D_i}$ then
            $O[j] \leftarrow P_i$
            $j \leftarrow j + 1$
    $t \leftarrow t + 1$
```

Using Algorithm 1, we can derive the vector O of expected reception order of N packets denoting the paths over which they will continue to be transmitted during the next $lcm(D_i \in \{D_1, D_2, ..., D_n\})$ duration of time as shown in (6.5). Now, from each path $P_i$ with corresponding path capacity $C_i$ and using the order in the vector O, we generate another vector $SEQ_{P_i}$ for each path $P_i$ which describes the packet sequence numbers that can be transmitted at every attempt of the scheduler to emit over the particular path. The method to derive $SEQ_{P_i}$ from O is shown in Algorithm 2.

**Algorithm 2** Deriving Packet Sequence Numbers to Transmit Per Path Using Expected Reception Order

```
$S_{max} \leftarrow 0$
for each $P_i \in \{P_1, P2, ..., P_n\}$ do
    $SEQ_{P_i} \leftarrow \mathrm{InitializeVector}()$
for each $P_i \in \{O_1, O_2, ..., O_{\sum_{i \in \{1,2,...,n\}} \frac{lcm(D_i)}{D_i}}\}$ do
    $SEQ_{P_i} \leftarrow \mathrm{Append}\,(SEQ_{P_i}, [S_{max}+1, S_{max}+C_i])$
    $S_{max} \leftarrow S_{max} + C_i$
```

Then we can easily schedule the next sequence of packets using Algorithm 3.

$$\begin{array}{ccc}
\text{Reception Time} & \text{Over } P_1 & \text{Over } P_2 \\
T_{10ms} & O(1) = P_1 & - \\
T_{20ms} & O(2) = P_1 & - \\
T_{30ms} & O(3) = P_1 & - \\
T_{40ms} & O(4) = P_1 & - \\
T_{50ms} & O(5) = P_1 & - \\
T_{60ms} & O(6) = P_1 & - \\
T_{70ms} & O(7) = P_1 & - \\
T_{80ms} & O(8) = P_1 & - \\
T_{90ms} & O(9) = P_1 & - \\
T_{100ms} & O(10) = P_1 & O(11) = P_2
\end{array} \tag{6.5}$$

**Algorithm 3** Transmission Based on Pre-calculated Sequence

```
$t \leftarrow 0$
while $t < lcm(D_i \in \{D_1, D_2, ..., D_n\})$ do
    for each $P_i \in \{P_1, P_2, ..., P_n\}$ do
        if $t \equiv 0 \pmod{D_i}$ then
            $\mathrm{Transmit}(P_i, SEQ_{P_i}[\frac{t}{D_i}])$
    $t \leftarrow t + 1$
```

With the same multi-path scenario and path parameters used in Table 6.1, we present another illustration of packet transmission using the delay aware scheduling in Table 6.2. First we derive $N = 25$ for this scenario using Equation 6.4. Algorithm 1 and 2 yield that packets $[1 \ldots 20]$ should be transmitted over $P_1$ while packets $[21 \ldots 25]$ should go over $P_2$. As can be seen in Table 6.2, since the packets sequences are carefully selected based on per path delay and emitted over the appropriate paths, the receiver's

| Xmit Time (ms) | Xmit over $P_1$ | Xmit over $P_2$ | Rcvd. Time (ms) | Rcvd. over $P_1$ | Rcvd. over $P_2$ | Good-put | Pkts in Rbuf |
|---|---|---|---|---|---|---|---|
| 0 | $1-2$ | $21-25$ | 10 | $1-2$ | none | $1-2$ | 0 |
| 10 | $3-4$ | none | 20 | $3-4$ | none | $3-4$ | 0 |
| 20 | $5-6$ | none | 30 | $5-6$ | none | $5-6$ | 0 |
| 30 | $7-8$ | none | 40 | $7-8$ | none | $7-8$ | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 90 | $19-20$ | none | 100 | $19-20$ | $21-25$ | $19-25$ | 0 |

Table 6.2: Delay Aware Packet Scheduling

buffer is never blocked in this non-oscillating deterministic scenario. In fact the buffer is always empty due to the proper in order arrival of the packets.

### 6.1.3 *Estimating Forward Delay for Delay Aware Scheduling*

In order to accurately estimate forward delay in CMT-SCTP, we propose a timestamp based method similar to the mechanism used for the RTT estimation in Datagram Congestion Control Protocol (DCCP) [13]. Although CMT-SCTP maintains its own estimation of RTT, in our previous work we have shown that RTT estimation in CMT-SCTP is severely degraded in presence of asymmetric transmission paths [31].

An accurate RTT estimation can easily be made without making significant modifications in the CMT-SCTP protocol. Our proposal to estimate RTT correctly includes adding some additional fields in the SCTP packet header namely *sender-timestamp* $(T_s)$ for data packets and *receiver-timestamp* $(T_r)$ and *time-elapsed* $(T_e)$ for SACK packets. An illustration of how the delay calculation may be performed using these timestamp fields is shown in Figure 6.3.

As shown in Figure 6.3, data packet $p_i$ is sent with sender-timestamp $T_{s1}$. It is received by the receiver and acknowledged at time $T_r$. If it takes $T_e$ amount of time for the receiver before sending the selective acknowledgement for data packet $p_i$, we can estimate the forward delay $D_{fwd}$ as shown in Equation 6.6 after having received the $SACK_{p_i}$.

$$D_{fwd} = T_r - T_e - T_{s1} \tag{6.6}$$

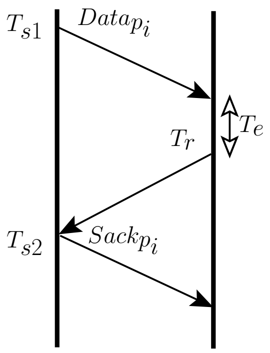


Figure 6.3: Timestamp based forward delay estimation

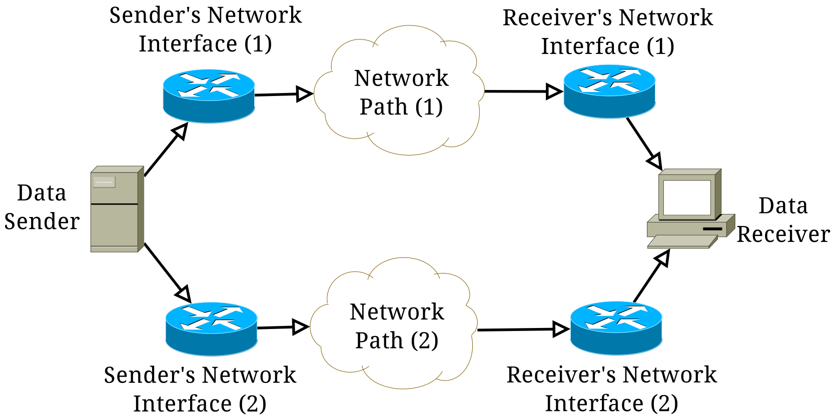

Sender's Network Interface (1)
Network Path (1)
Receiver's Network Interface (1)
Data Sender
Data Receiver
Network Path (2)
Sender's Network Interface (2)
Receiver's Network Interface (2)

Figure 6.4: Simulation topology

In the following section, we evaluate the improvements resulting from our proposal, by comparison with the baseline performance of the multi-path protocol using round robin scheduling.

## 6.2 experimental evaluation

To check the validity of our proposal from Section 6.1, we have implemented both the round-robin scheduling and delay-aware scheduling for multi-path data transfer in GNU's Matlab equivalent Octave tool [167].

A mathematical simulation model of a parametric packet scheduler was implemented with GNU Octave where the data-sender used two separate network paths to transmit data to the data-receiver with a network topology as shown in Figure 6.4. Network characteristic parameters such as path delay and available bandwidth over the paths could be varied as simulation input parameters with this implementation. Since our aim was to identify the maximum achievable network bandwidth for a given multipath topology and a packet scheduling method, varying network conditions and packet loss were omitted from the simulation. The implementation was able to

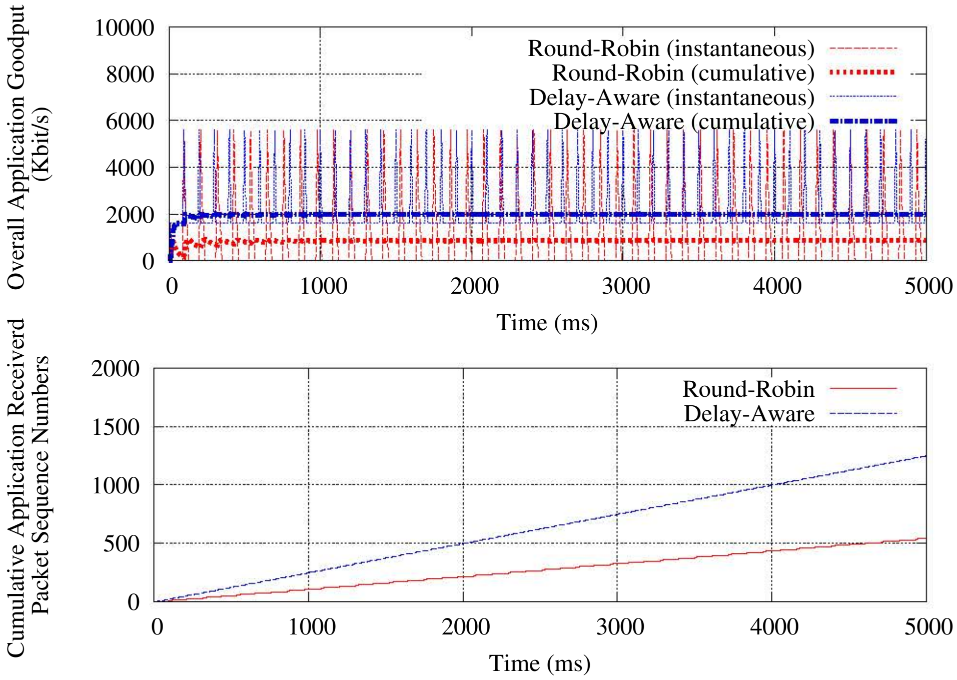


Figure 6.5: Comparison of round-robin and delay-aware scheduling

simulate both the round-robin and delay-aware packet scheduling methods. The round-robin scheduling implementation was also cross-validated with the simulation results from NS-2 [113]. In order to validate our results, we simulated the exact same network topology with NS-2 and cross-checked the results from GNU Octave simulation with the results from NS-2 simulation for round-robin packet scheduling, which is the default packet scheduling policy for vanilla CMT-SCTP.

The network topology used during the simulations was as shown in Figure 6.4. The path parameters are as per the example presented in Table 6.1, with path $P_1$ having $RTT_1 = 20ms$ and $B_1 = 1.6Mbit/s$; path $P_2$ with $RTT_2 = 200ms$ and $B_2 = 400Kbit/s$.

In Figure 6.5, we present a snapshot of the simulation performed with the same parameters presented in Table 6.1 and 6.2. We show cumulative packet sequence numbers received by the application and the application goodput resulting from aggregated data transfer on both paths. As can be seen, the performance of delay-aware scheduling is clearly much better than the round-robin scheduling both in terms of overall goodput and minimization of jitter experienced by the application.

In Figure 6.7, we present a comparison of the receiver's buffer usage and unacknowledged data packets on the flight. As can be observed, the delay aware scheduling clearly results in lower occupancy of the receiver's buffer, while also emitting more data packets which eventually leads to higher application goodput.

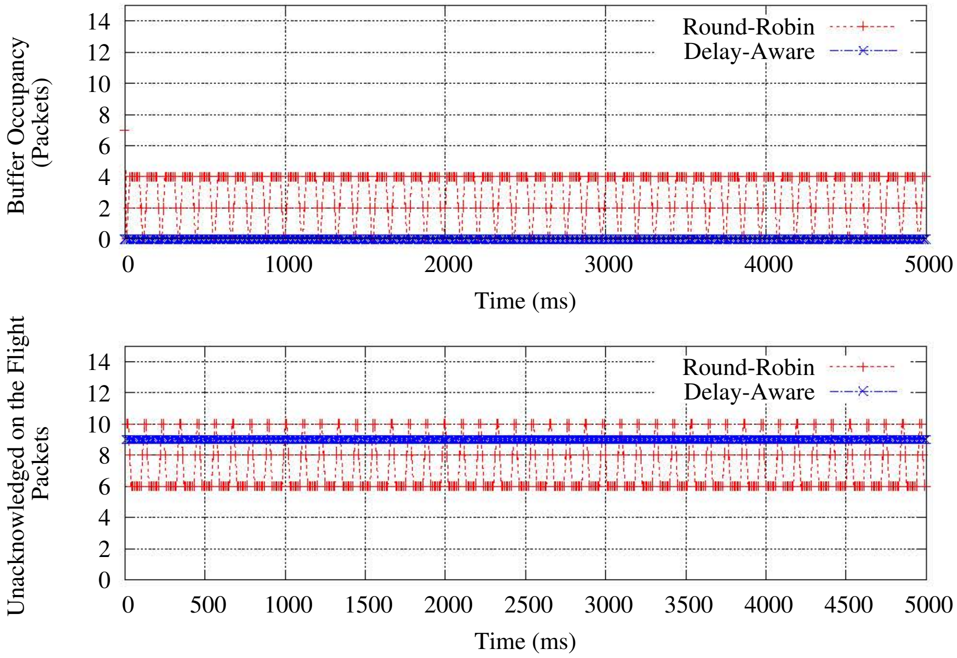


Figure 6.6: Comparison of receiver's buffer occupancy and unacknowldeged on the flight packets for round-robin and delay-aware scheduling

### 6.2.1 *Impact of Incorrect Delay Estimation on Delay Aware Scheduling*

We now present practical considerations related to the performance of our proposed scheme. The results presented in Figures 6.5 and 6.7 assume perfect estimation of the delay on both paths. We now present results for the case where there is an error in the estimated delay value(s), in Figures 6.8 and 6.9.

Figure 6.8 shows the goodput available to the application from the total data transfer on both paths. Baseline results are shown for the round-robin scheduling (bottom curve), the delay aware scheduling where there is no error in the delay estimation on both paths (top curve), and for the error in the delay over-estimation ranging from 10% to 100%. It can be observed that there is a solid gain by the delay aware scheduling mechanism, even for the case when there is a 100% error (twice the original delay) in the delay estimation.

Figure 6.9 presents the cumulative packet sequence numbers received by the application, for the first 5 seconds of the data transfer. Again, the bottom curve represents the result for the round-robin scheduling, and the top curve the delay aware scheduling where there is no error in the delay estimation on both paths. The middle curves represent results for the error in the delay estimation which is varied between 10% and 100%. Similarly to the application goodput results, it can be seen that a considerable error

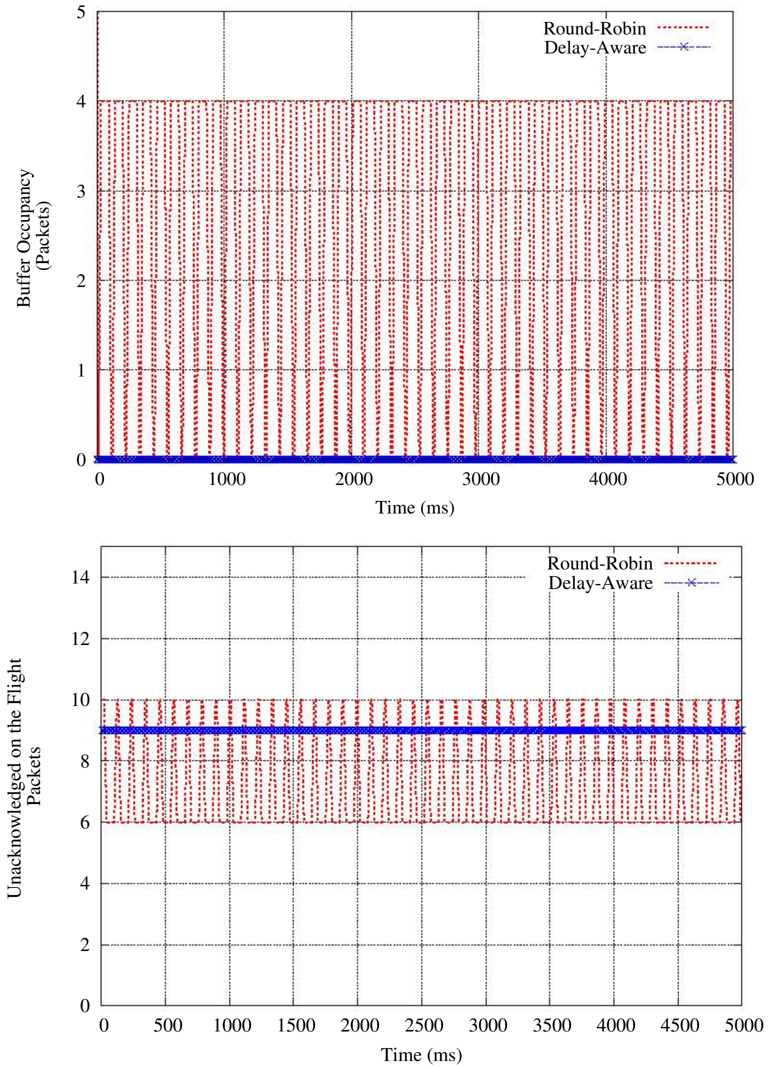


Figure 6.7: Comparison of receiver's buffer occupancy and unacknowledged on the flight packets for round-robin and delay-aware scheduling

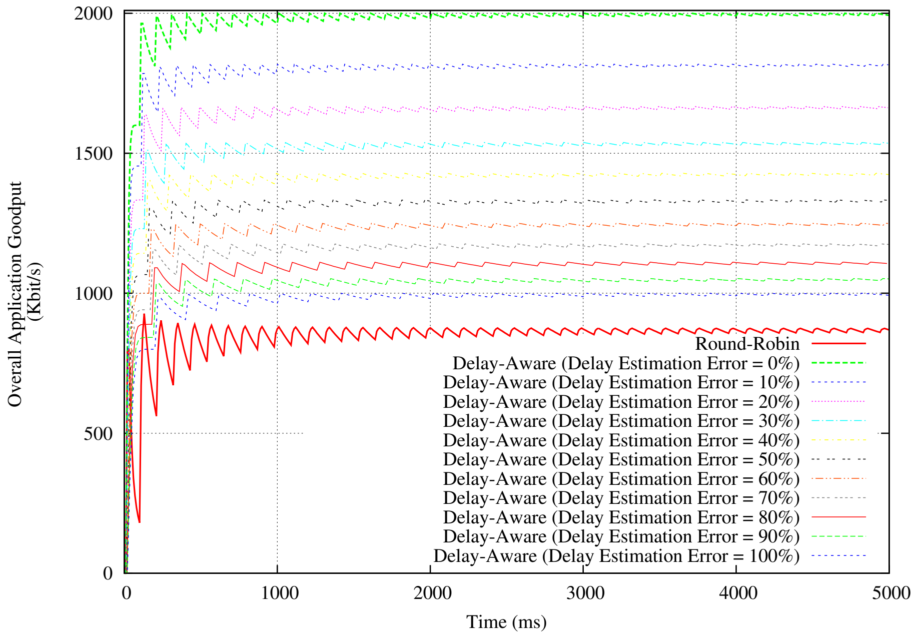


Figure 6.8: Impact of Incorrect Delay Estimation on Delay Aware Scheduling: Comparison of the overall application goodput

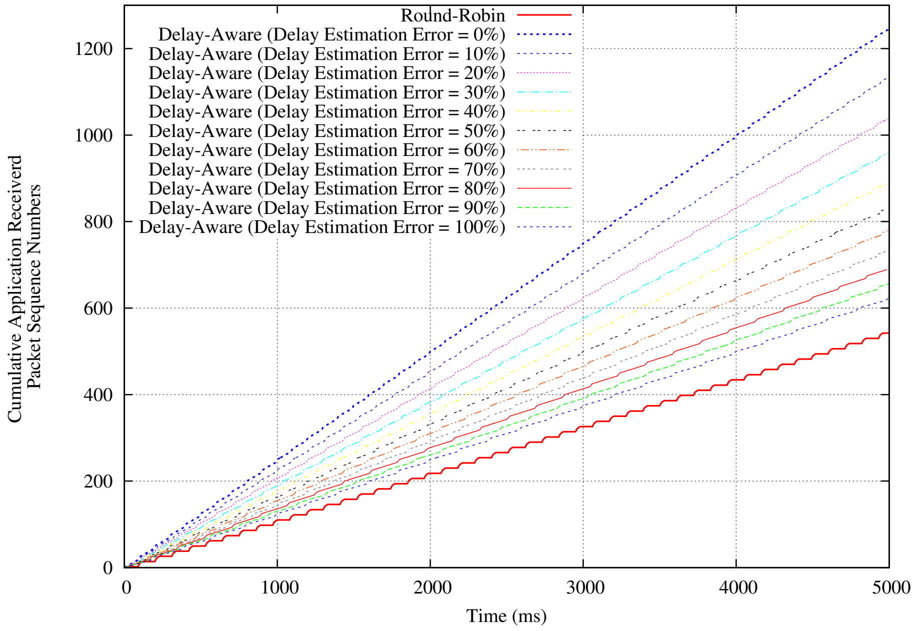


Figure 6.9: Impact of Incorrect Delay Estimation on Delay Aware Scheduling: Comparison

in estimating the delay on both path can be tolerated by the delay aware scheduling mechanism, while still providing an improvement compared to the round robin scheduling.

We note that the results for the occupancy of the receiver's buffer are omitted for the case of imperfect delay estimation, as they would show a similar trend as the comparison shown in Figures 6.8 and 6.9, i.e. that even a considerable error in the delay estimation by the transport protocol, used in the delay aware scheduling, still results in a solid improvement compared to the round robin scheduling mechanism.

## 6.3 CONCLUSION

Our current evaluation indicates that the delay-aware scheduling has significant potential for providing performance improvement over the traditional round-robin scheduling in asymmetric multi-path scenarios. As future work, we plan to implement our proposal first in NS-2 and later in FreeBSD's CMT-SCTP stack to evaluate the performance gain in realistic network conditions and address the related practical issues.

# 7 ECMT-SCTP: IMPROVING CMT-SCTP WITH ERASURE CODING OVER LOSSY LINKS

Performance of transport protocols on lossy links is a well researched topic, however there are only a few proposals which make use of the opportunities presented by the use of erasure coding within the transport layer, particularly in the multi-path transport protocol context. In this chapter, we investigate the improvements in the performance of multi-path SCTP transport protocol brought by the novel integration of the on-the-fly erasure code into the congestion control and reliability mechanisms of CMT-SCTP.

Compared to the existing work on improving performance of multi-path TCP and SCTP with various error correcting codes in a lossy environment, our key contributions presented in this Chapter are as follows.

We study the feasibility and the potential benefits of CMT-SCTP with erasure codes. We consider three different types of erasure codes, i.e. block codes, convolutional codes and on-the-fly erasure codes integrated within CMT-SCTP and we evaluate the performance of such a system for generic web applications using fully reliable CMT-SCTP and for video streaming using an equivalent of partially reliable CMT-SCTP.

We integrate three different types of erasure codes, i.e. block codes, convolutional codes and on-the-fly erasure codes within the transport layer and the evaluation of the resulting performance for generic web applications using fully reliable CMT-SCTP and for video streaming using non-reliable and an equivalent of partially reliable CMT-SCTP.

To further improve the performance, we propose a modification of the retransmission mechanism at the SCTP data sender with a variable retransmission delay ($\alpha$RTX), based on the type of error correction code. This includes a method to estimate the variable sender retransmission delay adjustment ($\alpha$RTX).

We finally present an analysis of CMT-SCTP performance with erasure codes based on the model from [168], validated against simulation results for a range of packet loss conditions.

We additionally provide an implementation of Explicit Congestion Notification (ECN) in NS-2 for both single-path SCTP and a the CMT-SCTP multi-path transport protocol.

The organization of the rest of the chapter is as follows: in Section 7.1, we elaborate our proposed integration of erasure codes with the transport layer and the adaptive retransmission scheme ($\alpha$RTX). In Section 7.2, we

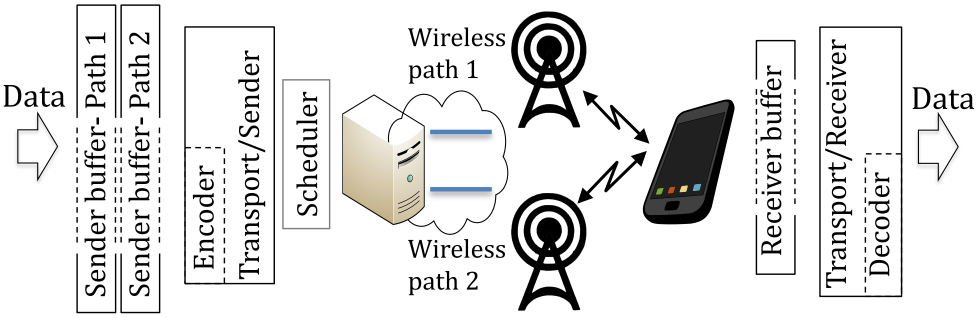


Figure 7.1: Architecture of the integrated multi-path transport and erasure coding

present an evaluation of our proposal, and demonstrate improvement both in terms for goodput for generic web applications and in terms of quality for video streaming. We provide an evaluation of eCMT-SCTP's performance validated against a multi-homed SCTP throughput model in Section 7.3. Finally in Section 7.4 present our conclusions.

## 7.1 OUR PROPOSAL

The changing conditions in the wireless environment, including short connectivity loss in mobility situations, result in a decreased performance of transport protocols and, consequently, in a lower quality of experience for the users. Erasure codes have been designed to handle lossy conditions and have been used in other proposals conjointly with transport protocols, e.g. in [169]. Our goal is to consider the potential benefits of this approach in situations where multiple wireless paths can be utilized for common mobile applications like web browsing and video streaming. To improve the multi-path transport performance in conditions of varying network loss, we propose eCMT-SCTP, which integrates erasure codes within the CMT-SCTP transport protocol. To further improve the performance of this integration, we propose a sender side modification to the packet retransmission scheme (aRTX).

In the CMT extension of SCTP [14], a single logical data connection can simultaneously use multiple physical network paths. In line with SCTP mechanism, congestion control uses a TCP-like window based algorithm, however a separate sender congestion control window (CWND) is allocated to each path. A single receiver window (RWND) is used for all paths. As per SCTP, CMT-SCTP includes partially and fully reliable transport with in-order or out of order packet delivery. For reliable transport with in-order delivery, which we focus on in this work, receiver uses selective acknowledgement (SACK) packets to acknowledge successful reception of data packets.

The proposed architecture to integrate erasure coding within the transport layer is shown in Figure 7.1. The central idea is to introduce redundancy inside the transmitted data flow at transport layer, applicable to all physical paths. Here, we show a multi-homed mobile with two wireless interfaces, connected to a server using two simultaneous network paths.

We assume that the imbalance between available paths (i.e. the difference in capacity and/or end-to-end delay) is mitigated either by intelligent scheduling at the sender side, as proposed in [33] or by a split buffer based solution proposed in [162]. In Figure 7.1, we show the advanced scheduler module, that allocates data to the different paths according to both the available window size and the difference in end-to-end delay on individual paths. Therefore, when evaluating the performance, we focus solely on evaluating the impact of lossy links.

In the proposed scheme, based on the specified encoding parameters, the sender encodes data packets and produces redundant packets as linear combinations of the data packets from the sender's buffer(s). The encoding process ensures that the data packets for which a redundant packet is produced, have not already been acknowledged by the receiver, i.e that redundant packets are generated only for the packets which are in the flight or were lost. A decoding at the receiver is attempted on arrival of every new packet. Once a successful decoding takes place, leading to recovery of a missing packet, a SACK packet is sent immediately to notify the sender. This is in contrast to the CMT-SCTP delayed SACK modification [14]. The (non-delayed) SACK allows the sender to immediately release buffered packets kept for retransmission leading to the increase of congestion window. For the case where the lost packet could not be recovered, the standard SCTP notification (gap report) is sent back to the sender.

We consider the use of on-the-fly convolutional codes and standard convolutional and block codes. The coding techniques are defined by the following parameters [170]. $k$ is the number of input data packets to be encoded, $n$ the number of output packets after encoding is performed and $m$ defines the memory length for the convolutional and on-the-fly erasure codes.

Both block and convolutional codes used in this chapter are systematic. A block code is defined by $(n, k)$. For an $(n, k, m)$ convolutional code, redundancy packet $R$ is created from previous $m$ input data packets from the sender buffer(s). On-the-fly erasure codes are a variation of convolutional codes, and TETRYS systematic erasure code [171] used in this chapter can also be represented by $(n, k, m)$. Note the code's memory size $m$ is variable and we can encode all available packets from the sender buffer into the encoded packet, as compared to the standard convolutional codes which have a fixed memory length.

### 7.1.1 *Congestion Control, Reliability and Erasure Codes*

In our proposal, redundancy packets are not considered by the congestion control and reliability mechanism of the transport protocol. Therefore, they do not occupy space in the sender's buffer, neither they are retransmitted if lost. It should also be noted that we do not introduce any additional decoding buffer in our proposed scheme. Rather, the same receiver's buffer is shared both by the data packets and the redundant packets. If the buffer is full, incoming data packets are prioritized over redundant packets. Similarly, if new redundant packets arrive, we perform a comparison between the redundant packet and the existing redundant packets in the buffer to quantify their utility given the current sequence of data packets in the buffer. The receiver-side buffer management algorithm when the buffer is full is presented in Algorithm 4.

**Algorithm 4** Receiver Buffer Management

```
function FIND_LEAST_ SIGNIFICANT
    LeastSig ← Null
    CumAck ← HighestAcknowledgedSequence
    for each Packet ← PacketsInRecvBuf do
        if Packet.Type = Data
        and NeededForDecoding(Packet) = False then
            LeastSig ← Packet
        if Packet.Type = Redundancy
        and EncodesPacketsBelow(CumAck) = True then
            LeastSig ← Packet
    if LeastSig = Null then
        LeastSig ← FindOldestRedundant(RecvBuf)
    if LeastSig = Null then
        Data ← FindOlderData(RecvBuf)
        if Data.Seq < CumAck then
            LeastSig ← Data
    if LeastSig = Null then
      return LeastSig
//Packet Reception Loop at Data Receiver
while Packet ← NewIncomingPackets do
    if RecvBuf is Full then
        LS ← Find_Least_Significant(Packet)
        if LS ≠ Null then
            RecvBuf[LS] ← Packet
        else
            Drop(Packet)
    else
        RecvBuf[Unoccupied] ← Packet
```

SCTP (and consequently CMT-SCTP) provides fully reliable, non-reliable [172] and partially reliable [94] transport options. Fully reliable SCTP provides reliability to an ongoing data flow by means of fast retransmissions and time-out retransmissions. If a packet is reported missing 4 times by the data receiver, SCTP sender will first attempt a fast recovery, while halving the congestion window. If the packet is still reported as missing, eventually the retransmission timer (RTO) will expire and a time-out retransmission will take place, consequently forcing SCTP into a slow-start phase for the corresponding path. During the initial evaluation of the integrated erasure codes, we have observed that while many of the lost packets were actually recovered by the erasure code, the same packets were also spuriously retransmitted by the sender. This was due to the independent operation of the erasure recovery (decoder) and the transport protocol's own retransmission based recovery mechanisms. To mitigate this, we propose an adaptive retransmission scheme which modifies the default fast and time-out mechanism of reliable SCTP with erasure codes.

### 7.1.2 *Adaptive Retransmission (aRTX)*

For the integrated erasure coding to be effectively utilised in reliable eCMT-SCTP, there has to be a sufficient delay in the SCTP retransmission mechanism to enable the lost packets to be recovered by the decoder, *i.e.* the system has to allow for the transmitting and receiving of the redundant packets which relate to the missing data packets. Therefore, our proposed adaptive retransmission (aRTX) scheme follows the below steps:

1. After a packet has been marked for fast retransmission, the retransmission is delayed at the sender by a timer value (in the number of packets) set to $\delta$, which ensures that the missing packet is not retransmitted at least until the transmission of the next redundant packet, which encodes the missing packet and until one more gap-report for the same missing packet is received.

2. Once a time-out is triggered, before performing a time-out recovery, the sender checks if an existing fast recovery is pending due to aRTX. If yes, the sender performs a fast recovery recalculating RTO for the fast retransmitted packets. Otherwise, standard time-out retransmission is performed.

The sender-side algorithm for the above steps is presented in Algorithm 5. As will be demonstrated by our simulation results, these rules allow room for the erasure code to perform packet recovery without the penalty in congestion window or flow control improving the overall performance further.

**Algorithm 5** Adaptive RTX ($a$RTX)

```
//Packet Transmission Loop at Data Sender
while Packet ← NewOutgoingPackets do
    if Packet.Type = DATA then
        Packet.$\delta$ ← Estimate_Delta(Packet)
//Packet Reception Loop at Data Sender
while Packet ← NewIncomingPackets do
    if Packet.Type = SACK then
        for each Missing in ReportedMissingPkts do
            if FastRtx is Due then
                if Missing.$\delta \neq 0$ then
                    Missing.$\delta$ ← (Missing.$\delta - \frac{s}{C}$)
                    Continue //aRTX Delay
                else
                    Fast_Recovery (Missing)
            if TimeoutRtx is Due then
                if Missing.$\delta \neq 0$ then
                    Fast_Recovery (Missing)
                else
                    Rto_Expiry(Missing)
```

### 7.1.3 *Method Used to Estimate δ*

For the case of reliable CMT-SCTP with erasure codes, the probability of spurious retransmissions can be approximated by the probability that the receiver will decode a lost packet after the receiver has transmitted a number of gap reports referring to the lost packet and the sender has retransmitted this packet. In $a$RTX, a sender will wait for $n$ packets to be transmitted including the redundant packet before triggering retransmission, therefore we have:

$$\delta \approx \left(n \times L \times \frac{1}{C}\right) + \frac{3RTT}{2} \qquad (7.1)$$

where $n$ is the number of sender transmitted packets after lost packet was transmitted, $L$ is the average packet size and $C$ is the link capacity. This estimation holds in the case of block code (because of the fixed size of a block) but not for convolutional and on-the-fly codes. In this case, our first attempts have shown that this estimation is complex to derive analytically. However, in the case of convolutional and on-the-fly codes, deriving an analytical estimation is not mandatory. Indeed, due to the SCTP sender congestion window progression, we know that it takes at least one RTT to transmit the whole window and at least one more RTT to receive 4 gap reports which trigger retransmission. Thus we propose to set $\delta$ to an upper bound corresponding to twice the RTT of the slowest path:

$$\delta = min(2 * RTT_i) \text{ with } i \in 0, 1, ..., n \tag{7.2}$$

where $RTT_i$ corresponds to the RTT of path $i$. As a first step, we propose to use this easy to implement upper bound. In future work we plan to refine this value following the model proposed in [173].

## 7.2 EVALUATION OF ECMT-SCTP OVER LOSSY LINKS

We have implemented our eCMT-SCTP proposal under existing CMT-SCTP contribution in NS-2 [113]. We have also implemented ECN for CMT-SCTP under NS-2 as per the IETF draft [105]. The network topology for our experiments follows the configuration shown in Figure 7.1, with a multi-homed mobile that has two wireless interfaces and is downloading a single data stream over two simultaneous network paths. Following the assumptions about path asymmetry being handled by one of the existing proposals [33, 162], we use the following parameters for all experiments: bandwidth and RTT for each of the paths are, respectively, $1Mbit/s$ and $100ms$. Both $Path1$ and $Path2$ have a uniform packet loss, which is varied from $1\% - 10\%$. Future work will consider fading channels with bursty losses, where on-the-fly coding scheme is known to perform well regardless of the configuration used [171]. All erasure codes used have equal redundancy of $10\%$, we used a $(10, 9)$ block code and $(10, 9, 20)$ convolutional code; with $m = 20$ being the initial value for on-the-fly convolutional code and the default size of the SCTP sender buffer, 43 packets, being the maximum $m$ limit (as $m$ varies for these codes). Therefore, out of $2Mbit/s$ aggregated available bandwidth over the two lossy paths, erasure coded CMT-SCTP had $1800Kbit/s$ effectively available, while the $200Kbit/s$ was dedicated to encoded redundant packets. Each of the experiments presented below was run $50$ times with a random seed and each for a duration of $300sec$.

### 7.2.1 *Fully Reliable eCMT-SCTP With Generic Web Traffic*

We first consider the performance of reliable transport, with generic web traffic *i.e.* web browsing or file download. Simulation results for the application goodput and the percentage of duplicate packets are shown in Figures 7.2 and 7.3. All graphs include the values of mean and standard deviation.

#### 7.2.1.1 *Single Stream Over Multipath without ECN*

We first present simulation results for a single-flow single-stream data transfer over two paths without support for ECN. Single-flow single-stream

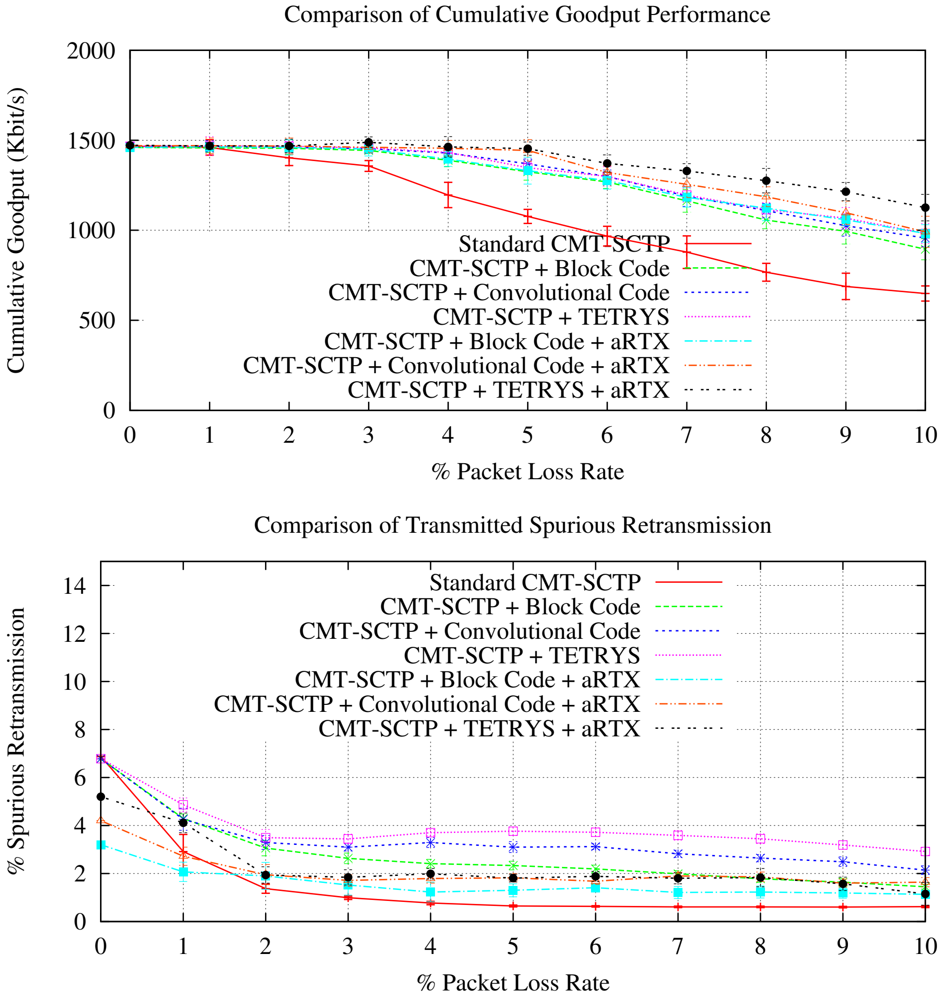


Figure 7.2: Performance comparison of standard CMT-SCTP and CMT-SCTP with erasure code and adaptive retransmission on ECN incapable network

represents a multi-path transfer where the overall data flow has a single sequence-number space. This is a challenging scenario for all current multi-path protocols, as packets arriving out of order occupy receiver's buffer eventually becoming a bottleneck blocking further data transmission. As can be seen in Figure 7.2, at around 2% packet loss we start achieving benefits of our erasure coding and aRTX proposals.

We can observe that the highest performance improvement is achieved with TETRYS and aRTX, followed by convolutional and block codes (also with aRTX). Although the absence of aRTX reduces the achievable performance gains, again TETRYS codes outperform convolutional and block codes in the simulated scenario. We note that, as shown in our previous work [31], CMT-SCTP fails to achieve the total aggregated throughput of 2Mbit/s due to SCTP's recommended default delayed SACK algorithm and receiver's buffer blocking even in the scenario which has no losses. This also applies to our eCMT-SCTP version, therefore restricting the total achievable goodput to around $1.5\mathrm{Mbit/s}$.

#### 7.2.1.2 *Single Stream Over Multipath with ECN*

We present simulation results of the same single-flow single-stream data transmission scenario, but over two ECN capable paths. As can be seen in Figure 7.3, although the ECN capable standard CMT-SCTP protocol demonstrates nearly the same pattern of degradation over lossy paths as shown in the non-ECN experiments, our proposed eCMT-SCTP outperforms the standard CMT-SCTP by a significant margin. It should be noted that in both of the scenarios TETRYS outperforms other erasure codes once the combined improvement ratio for goodput and reduction in spurious retransmission is considered. As shown in [171], TETRYS is actually expected to perform even better if the transport protocol itself does not introduce any bottleneck. As previously noted, single-flow single-stream data transfer over multiple paths is a challenging scenario for all currently proposed multi-path protocols, as any imbalance in individual paths leads to out of order packet arrival and eventual buffer blocking [162, 31]. Therefore, we have purposefully focused this study on the improvements erasure codes could bring to a scenario in which paths are symmetrical, as the main aim (and utility) of the proposed scheme is in improving the performance of erroneous lossy paths. Enhancement to multi-path protocols to overcome the path imbalance issue is one of the challenges we plan to study in future work.

### 7.2.2 *Equivalent Partially Reliable eCMT-SCTP With Video Streaming*

We now present the results for the performance of video streaming over the same simulation scenario from Figure 7.1. As CMT-SCTP (or SCTP) in NS-2 does not include the implementation of partial reliability as defined in [94], we use the fully reliable data transmission during simulations and introduce a delay constraint *i.e.* each packet that is delayed by more than $d_{max}$ is discarded by the streaming video playback application. Although this evaluation does include unnecessary retransmissions, we consider it a good first order approximation of the performance of partially reliable eCMT-SCTP. We choose $d_{max} = 150ms$ as defined by ITU-T G.114 [174] and the same way used in [175]. Similarly to the evaluation of fully reliable eCMT-SCTP, each experiment was run 50 times, with a random seed for the simulated loss, and with a $300sec$ duration of each run. We use the processed results of NS-2 simulations (with delayed packets filtered with $d_{max}$ as appropriate) as input to ITU-T G.1070 [176] recommended opinion model for evaluating video and telephony applications. G.1070 opinion model provides us with opinion scores as represented by $V_q$ in Equation 7.3,

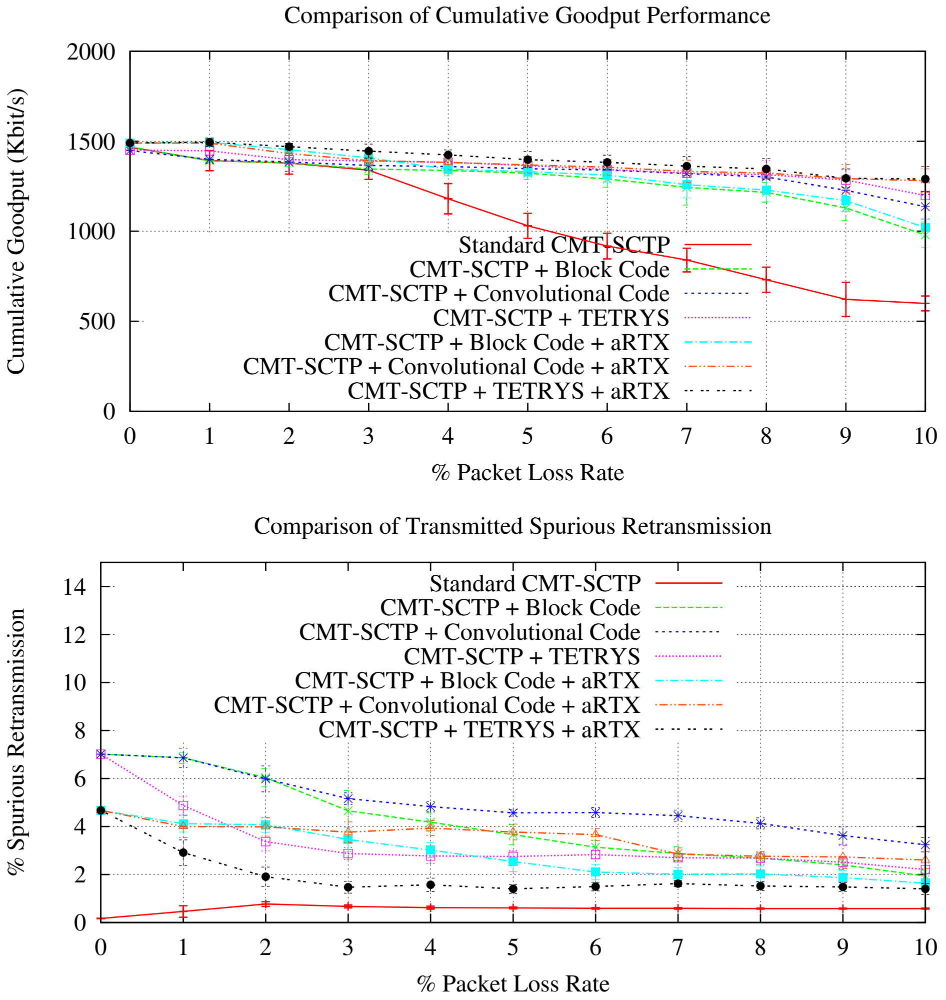


Figure 7.3: Performance comparison of standard CMT-SCTP and CMT-SCTP with erasure code and adaptive retransmission on ECN capable network

where $P_{plv}$ is the packet loss rate and $D_{P_{plv}}$ is the degree of video quality robustness due packet loss in the application layer. Basic video quality $I_c$ is calculated as shown in Equation 7.4. Video quality $V_q$ with no packet loss in the network is expressed by Equation 7.5. $b$ in these equations represents bit-rate of the encoded video and coefficients $v3$, $v4$ and $v5$ depends on the type of video codec used, video playback display format, intervals in video key-frames and video playback display size as per ITU-T recommendations in G.1070.

$$V_q = 1 + I_c . e^{\frac{P_{plv}}{D_{P_{plv}}}} \tag{7.3}$$

$$I_c = v_3 . \left(1 - \frac{1}{1 + \left(\frac{b}{v4}\right)^{v5}}\right) \tag{7.4}$$

$$V_q = 1 + v_3 . \left(1 - \frac{1}{1 + \left(\frac{b}{v4}\right)^{v5}}\right) \tag{7.5}$$

In Figure 7.4, first we show the packet loss rate as experienced by the video playback application with respect to the late incoming packets with $d_{max} = 150ms$. Then we use these values to calculate the video quality evaluation opinion metric defined by the ITU-T G.1070 model [176]. Figure 7.5 shows the perceived video quality represented by Mean Opinion Score (MOS) for a range of packet loss ratios on the network link. For video quality evaluation, we have used standard MPEG4 video codec with QVGA resolution (Quarter VGA, $320 \times 240$ pixels), bit-rate $b = 1000kbit/s$, frame-rate $30fps$, key-frame interval of $1fps$ and with a video playback display size of $4.2inch$. Provisional values as defined in ITU-T G.1070 are used for the rest of the parameters including coefficients $v3$, $v4$ and $v5$. In Figure 7.5, we can observe that the perceived video quality as defined by ITU-T G.1070 is greatly improved by the erasure codes compared to the standard CMT-SCTP when there is packet loss in the network.

We have also analysed the video application's play-out buffer evolution, *i.e.* the level of jitter introduced by the erasure coding mechanism and aRTX with respect to the lossy network paths. We compare the received jitter for standard CMT-SCTP and eCMT-SCTP in Figure 7.4. We can observe that jitter experienced by the video application notably decreases with the introduction of our erasure coding schemes due to early packet recovery without retransmission.

## 7.3 PERFORMANCE VALIDATION

In this section, we present the performance validation of the fully reliable eCMT-SCTP by using the multi-homed SCTP analytical model from [168].

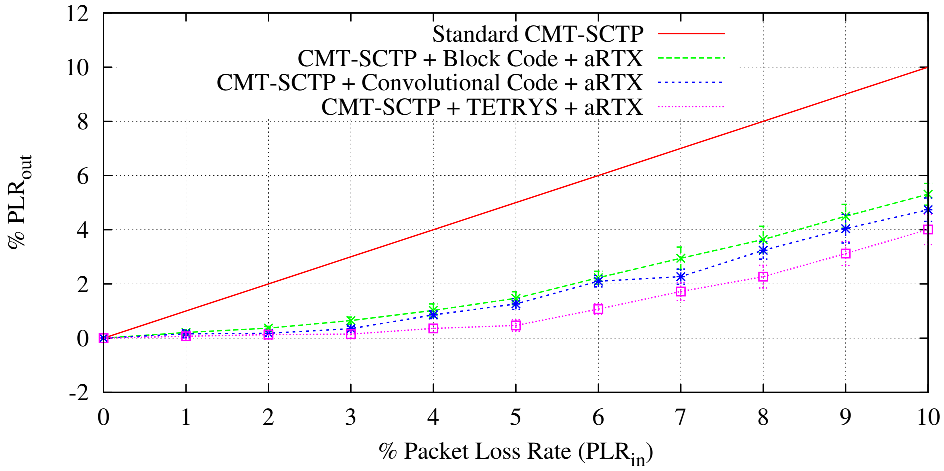


Figure 7.4: Packet error rate ($PLR_{out}$) as experienced by the video playback application with maximum tolerable delay for the incoming packets of $d_{max} = 150ms$

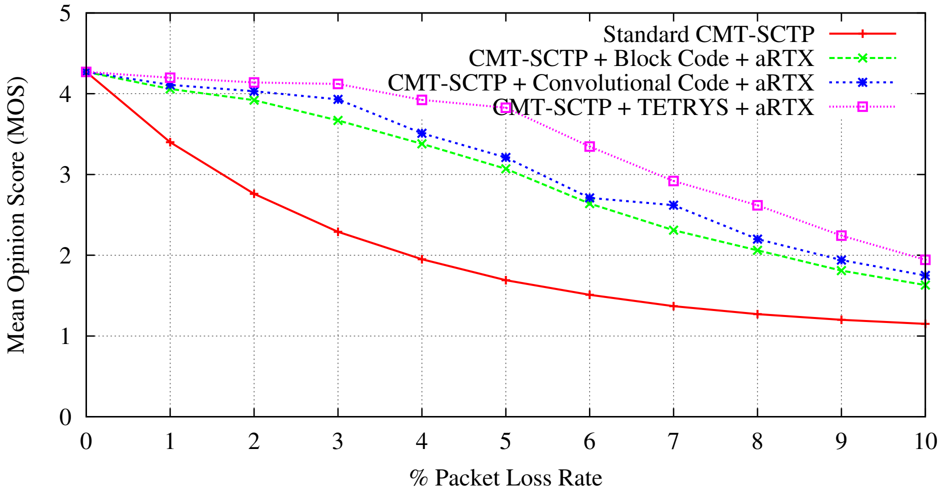


Figure 7.5: Mean Opinion Score (MOS) computed by ITU-T G.1070 model

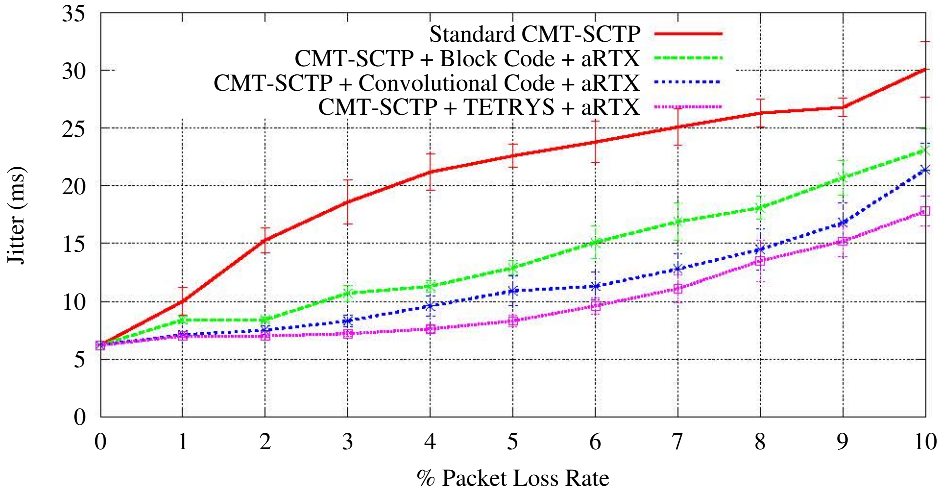


Figure 7.6: Jitter as experienced by the video playback application

The estimated aggregated goodput is calculated as shown by the Equation 7.6 with respect to the total usable data capacity of $1800\mathtt{kbit/s}$ and varying loss of $1\% - 10\%$ in our experiments. As can be seen in Equation 7.6, this model is function of packet loss expressed by $p_j$ for each of the paths in a multi-path scenario. In Equation 7.6, $W_j$ is the congestion window of the corresponding paths, $RTT_j^r$ mean RTT for retransmission, $RP_j$ is the mean number of packets transmitted via alternative path, $W_j^r$ is the mean congestion window during retransmission and $W_{Max}^j$ is the maximum window size which is the size of the receiver's window in our case. $Q_j$ in Equation 7.6 is calculated as shown in Equation 7.7. $RTT_j^r$ and $W_j^r$ are calculated as Equation 7.8 and 7.9 respectively. Calculation of $RP_j$ is shown by the Equation 7.10 and $W_j$ is calculated as 7.11.

For selected erasure codes, we measure the corresponding packet loss rate at the decoder output in NS-2 experiments that utilise eCMT-SCTP with unreliable data transmission. This allows us to capture the true packet loss rate ($PLR_{out}$) in presence of the erasure codes as experienced by the transport layer. Figure 7.7 shows the input to output packet loss rates for block, convolutional and on the fly convolutional codes for a varying input packet loss rate of $1\% - 10\%$. We use these values in Equation 7.6 for eCMT-SCTP goodput calculation. Figure 7.8 shows the resulting eCMT-SCTP throughput estimates, compared to the NS-2 simulation results for fully reliable eCMT-SCTP. We only show a subset of the results, with modelling of aRTX retransmission mechanism left for future study. We can observe a closer match between the model and simulation results as the packet loss in the network increases, while a notable fixed mismatch remains in case of no and minimum packet loss. Our analysis shows that this mismatch is primarily contributed by the implications of receiver's buffer blocking and imprecise blind round-robin packet scheduling for CMT-SCTP in NS-2 simulations which the model in [168] doesn't take into account.

$$B_j(p_j) = \min\left(\frac{W^j_{\text{Max}}}{RTT_j}, \frac{Q_j \cdot (1.5W_j - 2) + 1.25W_j + (1-p_j)/(p_j)}{\begin{array}{r}Q_j \cdot To + RTT^r_j(1/2 + RP_j/W^r_j) + \\ RTT_j Q_j \cdot \log 2.5(W_j+2)/2 + (W_j-2)/2\end{array}}\right) \tag{7.6}$$

$$Q(w) = \min\left(1, \frac{\left(1-(1-p)^4\right)\left[1+(1-p)^4\left[1-(1-p)^{w-3}\right]\right]}{1-(1-p)^w}\right) \tag{7.7}$$

$$E[RTT^r_j] = \sum_{l=1, l\neq j}^{N} \frac{\frac{1}{p_l}}{\sum_{m=1,m\neq j}^{N}\frac{1}{p_m}} \times RTT_l \tag{7.8}$$

$$E[W^r_j] = \sum_{l=1, l\neq j}^{N} \frac{\frac{1}{p_l}}{\sum_{m=1,m\neq j}^{N}\frac{1}{p_m}} \times E[W_l] \tag{7.9}$$

$$RP(w) = \frac{(3-w).p^3 + (4w-11).p^2) + (14-6w).p + (4w-6)}{-p^3 + 4p^2 - 6p + 4} \tag{7.10}$$

$$E[W_j] = \frac{4}{3b} + \sqrt{\frac{8}{3bp} + \left(\frac{4}{3b}\right)^2} \tag{7.11}$$

## 7.4 CONCLUSION

From the presented results above, our erasure coded schemes clearly start to demonstrate benefits for $2\% - 3\%$ packet loss in the network. Our adaptive RTX scheme helps improve the performance of evaluated erasure codes further both by improving overall goodput and reducing spuriously retransmitted duplicate data packets saving useful bandwidth. Also, memory constrained mobile devices should be able to benefit from this eCMT-SCTP scheme, as our proposal does not introduce any additional buffering but

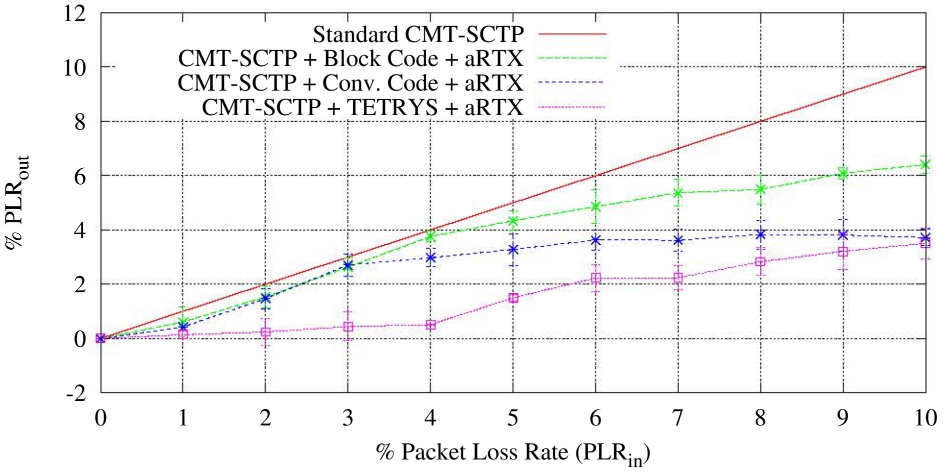


Figure 7.7: Packet error rate ($\mathsf{PLR_{out}}$) experienced by CMT-SCTP

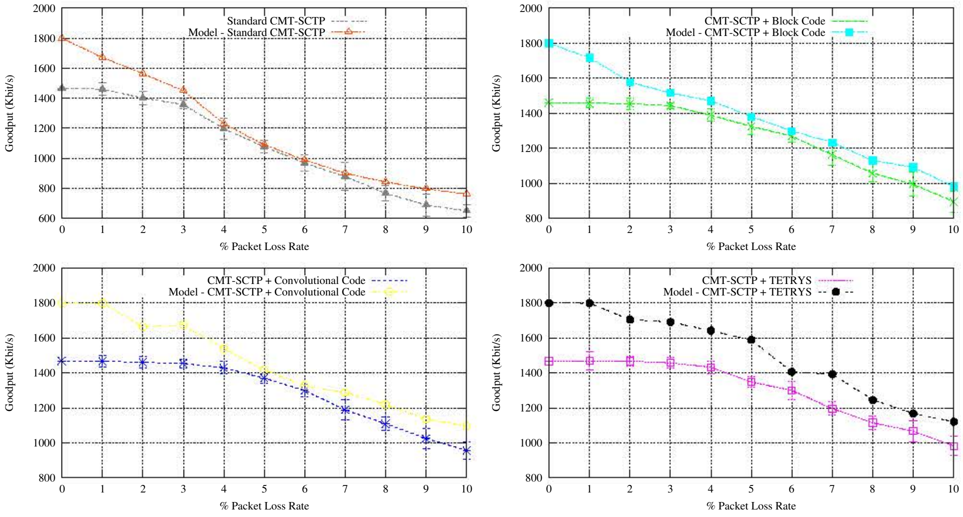


Figure 7.8: Performance validation of eCMT-SCTP

innovatively shares the already existing buffer within the transport protocol. We plan to extend this work by integrating context based variable redundancy for bursty network loss in asymmetric multi-path networking scenarios. Due to space constraints, we omit the evaluation of fairness to competing flows. We note that using ECN is beneficial to fairness when standard and eCMT-SCTP are concurrently used.

# 8 TOOLS DEVELOPED

## 8.1 CROSS-LAYER VIDEO STREAMING WITH DCCP

We present Xstream-x264: a real-time cross-layer video streaming technique implemented within a well known open-source H.264 video encoder tool x264 for *on-line* video quality evaluation experiments. Xstream-x264 uses the transport protocol provided indication of the available data rate for corresponding adjustments in the video encoder. We discuss the design, implementation and the quality evaluation methodology utilized with our tool. We demonstrate via experimental results that the streaming video quality greatly improves with the presented cross-layer approach both in terms of lost frame count and the objective video quality metrics Peak Signal to Noise Ratio (PSNR).

### 8.1.1 *Design of the Cross Layer Video Tool*

H.264/MPEG-4 is a widely used standard for video compression today [177]. H.264 video, when streamed over congested communication channels, will incur a loss of quality if video frames are lost during transmission [178]. For real-time multimedia streaming, this creates a bigger problem as in-time delivery is crucial and reliability achieved using retransmissions may not satisfy the in-time requirement. To minimize losses and maximize end users' viewing experience, we propose a simple and efficient cross layer technique which adjusts the parameters of the ongoing video encoding process based on external information (e. g. throughput, loss information about the link found in transport protocol).

The most common H.264 parameters specified for the encoding process are either a user-specified constant quality or a constant bit-rate (CBR) [179]. We note that CBR encoding results in all frames being encoded with the same value of the maximum target bit rate, while constant quality, which effectively is VBR encoding will result in a data rate which is proportional to the degree of complexity and motion in the video. In our proposed technique, the encoder effectively encodes with both a variable quality and a variable rate by following an externally provided rate based on cross layer information. The encoding process changes the rate when a change in the available bandwidth is detected, due to loss or congestion. This allows the application to conform to the available bandwidth on the link resulting in a lower number of frame losses and provides an improved viewing experience.

We have implemented the proposed technique in a tool, by modifying an open source H.264 encoder x264 [179]. The implementation also includes a real time encoding and streaming capability and additional features which aid the evaluation of video quality when the received video has frame losses.

The functional components of our Xstream-x264 tool are shown in Figure 8.1.

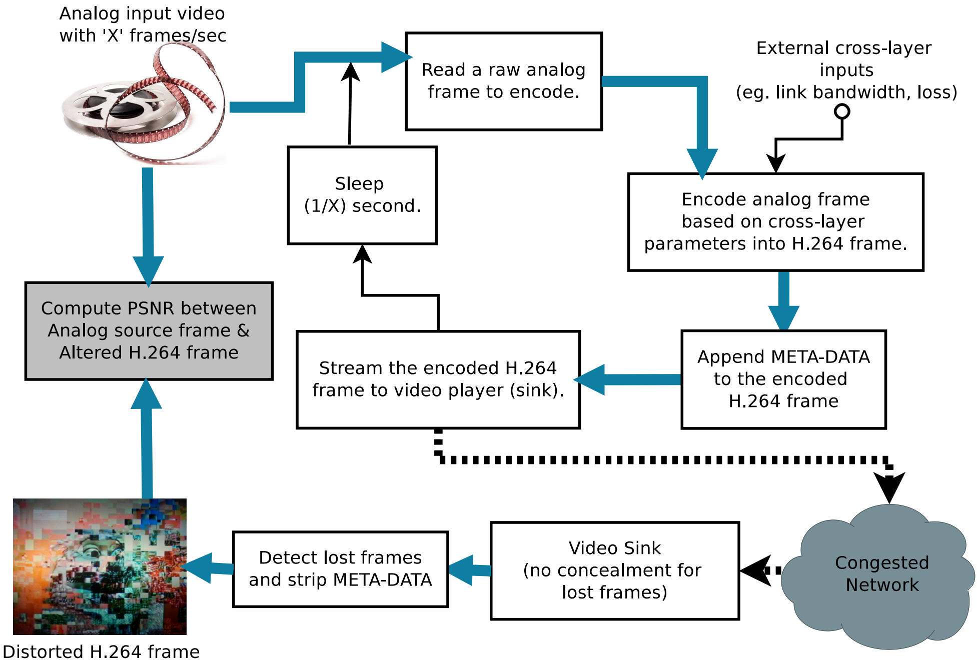

Analog input video
with 'X' frames/sec
Read a raw analog frame to encode.
External cross-layer inputs (eg. link bandwidth, loss)
Sleep (1/X) second.
Encode analog frame based on cross-layer parameters into H.264 frame.
Compute PSNR between Analog source frame & Altered H.264 frame
Stream the encoded H.264 frame to video player (sink).
Append META-DATA to the encoded H.264 frame
Detect lost frames and strip META-DATA
Video Sink (no concealment for lost frames)
Congested Network
Distorted H.264 frame

Figure 8.1: System Model of Xstream-x264

The analog video source is encoded into H.264 frames based on throughput or link capacity information provided transport protocol (although other external parameters may also be used from other layers in OSI network stack). Meta-data, including frame sequence number and fragmentation sequence number is appended to the encoded frame (a total of 4 bytes per frame), which is then streamed with timing corresponding to the source video. The H.264 video is sent over the link of choice and captured by the video sink. This video is passed through the loss detection module we have also implemented, followed by the computation of video quality.

To evaluate the received video quality, we use Peak Signal to Noise Ratio (PSNR), the most widely used objective video quality metric standardized by ITU [180]. Although PSNR may not always accurately estimate the subjective video quality due to non-linear nature of the human visual assessment, studies have shown that it does provide a reasonable estimate of the subjective assessment if it is applied over the same video content under the same environment [181]. We use the *Compare* component of ImageMagick [182] tool-set for PSNR computation, in conjunction with our frame loss detection module. To enable PSNR computation using two streams which have an unequal number of video frames, for any detected lost frame, we insert a blank (all white) frame. The existing playback concealment techniques [183] can still be utilized in addition to our proposal and should further improve the results in terms of visual perception. We have however chosen not to use them in our experiments, in order to more clearly show the benefits of our proposal.

Our tool can currently be used with either UDP/RTP or DCCP with Congestion Control ID 3 (CCID3) [24]. DCCP was targeted as it is a good transport protocol candidate for future multimedia applications, with CCID3 specifically designed to work with data and video transmission [24].

### 8.1.2 *Experiments and Analysis*

To demonstrate the capability of Xstream-x264, in this section we present experimental results showing the effects of congestion. The simple network topology used in the experiments includes a source (video sender) and a sink (video receiver), connected by a router. Congestion is emulated in the router by using the Linux network emulator *NetEm* rather than additional traffic streams. The nominal link bandwidth is 1 Mbit/s and the Round-Trip-Time (RTT) is 10ms for all the experiments. All experiments are of 42 seconds duration and the artificial congestion is introduced by reducing the link bandwidth to 700 Kbit/s after 15 seconds; after the 30th second, the link bandwidth is further decreased to 400 Kbit/s. The Maximum Transmission Unit (MTU) of the network is set to 1400 bytes. If an encoded frame is larger than the MTU, the video frame is fragmented and transmitted in subsequent packets without any intermediate frame delay. Otherwise, the encoded frames are transmitted in real time, with respect to the intra-frame delay of the raw input video. The analog input video used in our experiment is a 41.66 seconds, 1000 frames long scene from a video clip of an action movie with a very large amount of motion. The input video plays at 23.976 frames per second (FPS).

We compare the calculated PSNR values for our proposed cross layer mechanism and a CBR encoded video stream, transmitted over the congested link. We use DCCP transport protocol to provide the cross-layer information about the link to the video application, we also use DCCP for the CBR experiments, with a constant bit rate (CBR) of 1 Mbit/s. To present a fair evaluation of how our proposed method utilizes the rate information provided by DCCP, we also perform experiments for an ideal scenario where the application knows the available rate on the link and changes the encoding parameters according to this presumed knowledge. The resulting PSNR values are shown in Figure 8.2, together with the congestion pattern used in the experiments. As can be expected, CBR (1 Mbit/s), which is not well suited to the available capacity of the link, experiences significant losses in the congested period and as a consequence, DCCP-CCID3 (which has a loss and RTT based congestion control mechanism [24]) severely restricts the rate available to the application. On the other hand, our proposed cross-layer VBR technique attempts to follow the available rate as advised by the transport protocol, thereby the result is much lower frame loss. The only loss is due to the frequency of DCCP rate change and the limitation of the application to only change the encoding rate on a per-frame basis. The ideal case where the application knows the exact rate rather than the DCCP information about the rate shows the highest PSNR.

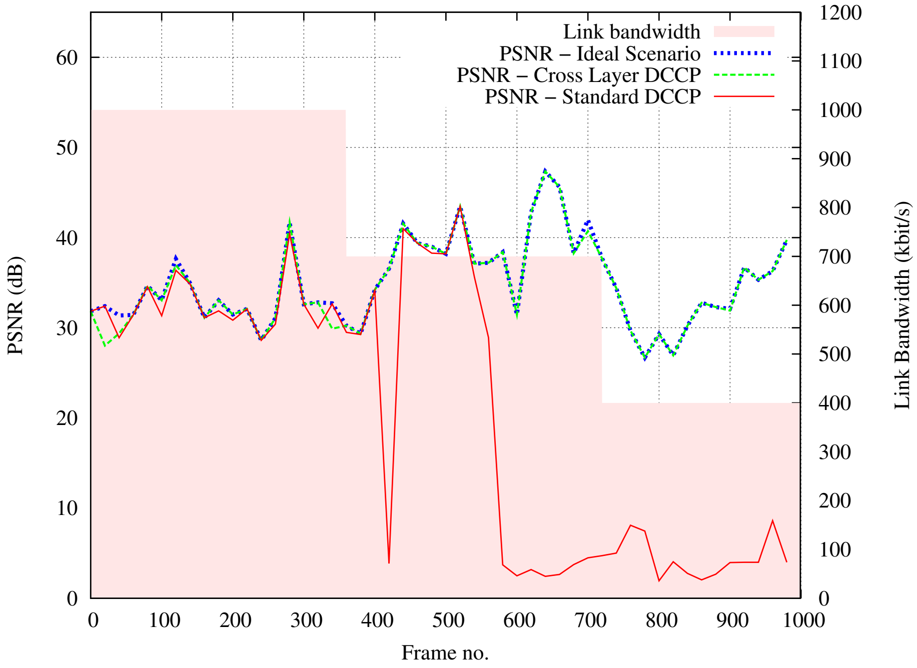


Figure 8.2: Comparison of PSNR for an ideal scenario, proposed Cross-layer DCCP and standard DCCP streaming.

The analysis of the PSNR values for the for ideal scenario, proposed cross layer DCCP (averaged over all possible initial rates from 100 Kbit/s to 1 Mbit/s) and standard DCCP is shown in Table 8.1. We present the mean and coefficient of variation (CV, ratio of standard deviation to mean) for the full experiment and, separately, for the three congestion regions.

To further demonstrate the advantages of the adaptive mechanism, we also show the number of lost frames from the same experiments in Table 8.2. Please note that the ideal scenario has no losses, therefore it is omitted.

Frame losses are separated into losses for the index (I), predictive (P) and bi-directional predictive (B) frames. In H.264, the I frames provide reference and necessary decoding information for the P and B frames and therefore lost I frames have a greater impact on quality. As can be seen in the tables, by adapting to our cross layer technique, not only we have reduced the overall number of lost frames, we have also cut down the number of lost I frames which guarantees improved video playback for the viewer.

| | | 1 Mbit/s | | 700 Kbit/s | | 400 Kbit/s | |
|---|---|---|---|---|---|---|---|
| | Overall Mean (dB) | Mean (dB) | CV (dB) | Mean (dB) | CV (dB) | Mean (dB) | CV (dB) |
| Ideal | 35.20 | 33.60 | 0.08 | 38.68 | 0.11 | 32.77 | 0.11 |
| Std. | 19.83 | 28.95 | 0.36 | 22.54 | 0.72 | 4.63 | 0.55 |
| Cross layer | 33.62 | 30.96 | 0.27 | 37.30 | 0.20 | 32.28 | 0.16 |

Table 8.1: Detailed analysis of PSNR from Figure 8.2. for the standard and cross-layer DCCP approaches.

| Enc. rate (Kbit/s) | % Total loss | % frame loss (I) | % frame loss (P) | % frame loss (B) |
|---|---|---|---|---|
| Stdandard | 47.75 | 0.60 | 25.13 | 21.82 |
| Cross-layer | 6.21 | 0.20 | 2.30 | 3.70 |

Table 8.2: % Frame Loss for Cross Layer DCCP and Standard DCCP (1 Mbit/s CBR).

We further extend our work by emulating the standard YUV video sequences provided by Xiph Foundation [184] and Arizona State University [185] for multimedia research. Videos of different resolution and motion complexity are chosen for cross layer based DCCP and standard DCCP streaming. All videos are encoded at 23.976 frames per sec (FPS). Standard DCCP streaming is performed at 1 Mbit/sec. Emulations are performed under the same congestion environment shown in Figure 8.2. Performance of cross layer DCCP using Xstream-x264 and standard DCCP streaming in terms of number of lost frames are presented in Table 8.3. As can be seen in Table 8.3, cross-layer based DCCP streaming with Xstream-x264 significantly reduces number of lost frames and shows notable improvement in all cases.

In Table 8.4, we present detailed analysis of lost I, P and B frames for the experiments with standard YUV video sequences presented in Table 8.3. Cross-layer based streaming significantly reduces the number dropped

| Video File | Resolution | Motion Complexity | % Lost Frames (Std.) | % Lost Frames (Cross-layer) |
|---|---|---|---|---|
| Foreman | 352 x 288 | High | 34.2 | 9.8 |
| Football | 352 x 288 | High | 37.1 | 9.2 |
| Stefan | 352 x 264 | High | 37 | 9.4 |
| Highway | 352 x 288 | High | 33.2 | 8.6 |
| Bus | 352 x 288 | High | 34.3 | 8.8 |
| Coastguard | 352 x 288 | High | 48.4 | 9.7 |
| Carphone | 176 x 144 | Medium | 17.3 | 5.1 |
| Mobile | 352 x 288 | Medium | 31.6 | 7.4 |
| Paris | 352 x 288 | Medium | 39.6 | 7.6 |
| Suzie | 176 x 144 | Medium | 23.1 | 4.1 |
| Akiyo | 352 x 288 | Low | 30 | 6.4 |
| Bridge (close) | 352 x 288 | Low | 23.7 | 1.1 |
| Bridge | 352 x 288 | Low | 30.6 | 1.1 |
| Container | 352 x 288 | Low | 54.9 | 14 |
| News | 352 x 288 | Low | 42.6 | 11.3 |
| Hall | 352 x 288 | Low | 29.1 | 6.9 |

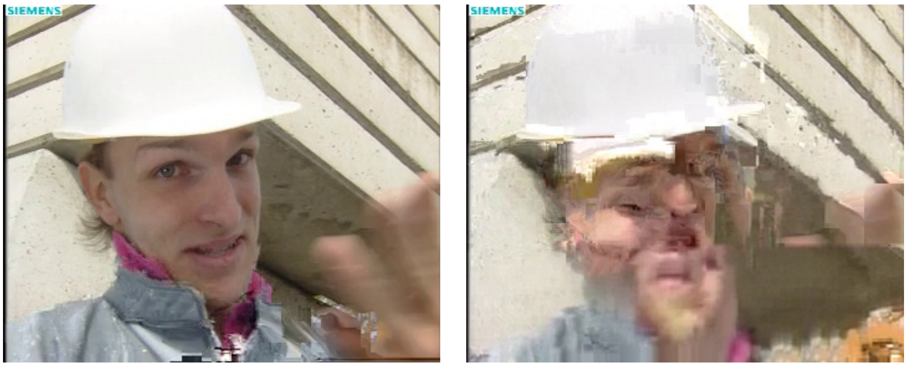
Table 8.3: % Frame loss of Cross Layer DCCP and Standard DCCP (1 Mbit/s CBR) for standard YUV video sequences.

I frames which substantially helps improve streaming video quality experience in a congested network environment.

We illustrate the visual quality difference in Figure 8.3 by comparison of a selected video frame from two sources: as decoded from the stream transmitted and received by our proposed approach and decoded from the 1 Mbit/s CBR received stream. It can be observed that the image in Figure 8.3b appears significantly more distorted than the image from Figure 8.3a.

### 8.1.3 *Discussion*

We have proposed and implemented a novel cross layer based video encoding and streaming technique and demonstrated it's feasibility by Xstream-x264 based implementation. Our experimental results show that cross-layer based streaming significantly outperforms the traditional streaming method



(a) DCCP Cross-layer

(b) Standard DCCP

Figure 8.3: Comparison of visual frame quality.

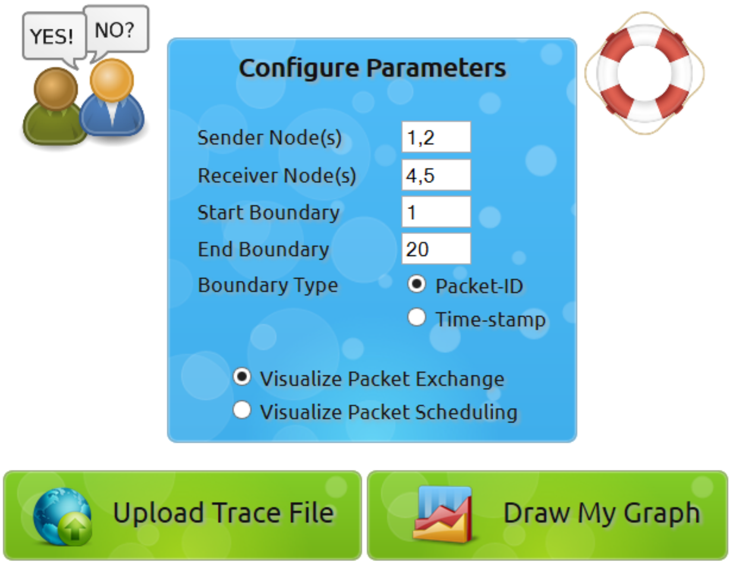
using DCCP transport protocol. As future work, we plan to integrate RTP transport protocol with our tool and make it available to a broader range of audience experimenting with multimedia encoding and streaming.

## 8.2 PACKET VISUALIZATION TOOL - "PACKETVIZ"

Using a combination of Python programming language [186], Scalable Vector Graphics (SVG) [187] and modern HTML5 [188] technologies, we developed a packet visualization tool called "PacketViz". The aim of PacketViz was to assist us in critically investigating how packet scheduling is inferred in single-path and multi-path transport protocols, where the hidden scheduling bottlenecks are and overall to find out how packet scheduling can even be improved.

PacketViz is currently capable of taking NS-2 trace files from TCP, UDP, DCCP, TFRC, SCTP and CMT-SCTP simulations. Based on the input trace file and some preferential user-specified parameters such as simulated network node numbers, packet ID or time-stamp (as shown in Figure 8.4), PacketViz is able to generate both packet scheduling graphs such as Figure 8.5 or packet timeline graph as shown in Figure 8.6.

We found PacketViz immensely useful during investigating, simulating, proposing and improving the delay aware packet scheduling for CMT-SCTP described in Chapter 6. Besides being an useful debugging tool, we believe PacketViz could also be used as great learning tool for the students to visually analyze the behavior of data transfer in the network.



Configure Parameters

Sender Node(s) 1,2

Receiver Node(s) 4,5

Start Boundary 1

End Boundary 20

Boundary Type ● Packet-ID ○ Time-stamp

● Visualize Packet Exchange

○ Visualize Packet Scheduling

Figure 8.4: PacketViz User Interface

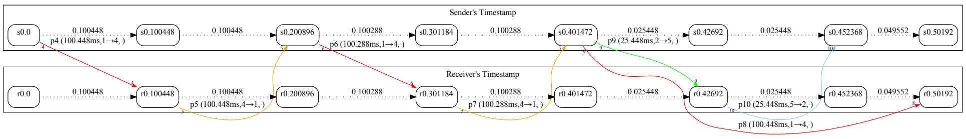


Figure 8.5: Graph based packet exchange visualization generated using PacketViz

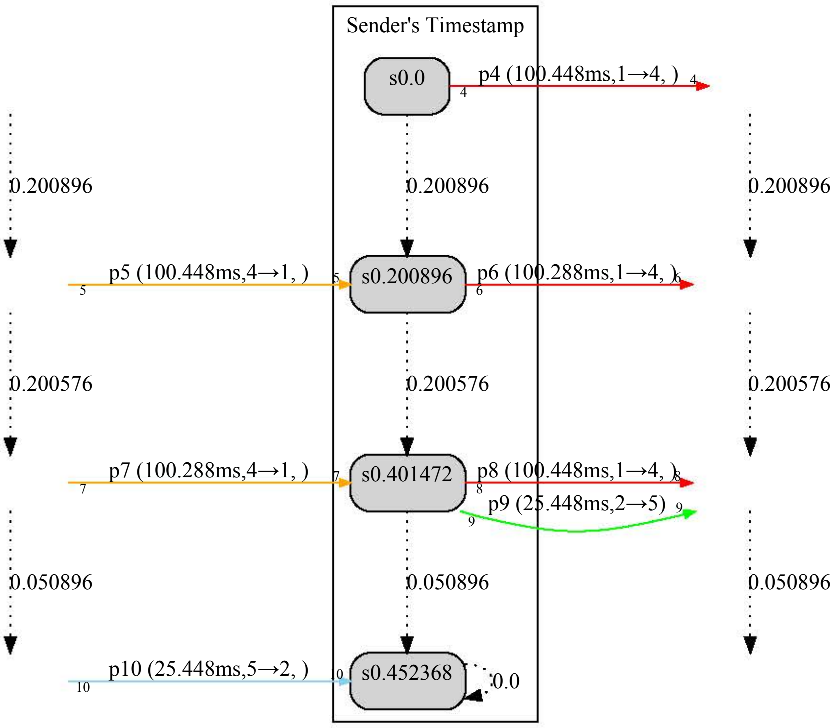


Figure 8.6: Graph based packet timeline visualization generated using PktViz

| Video File | Encoding Method | % I Frame Loss | % P Frame Loss | % B Frame Loss | % Total Loss |
|---|---|---|---|---|---|
| Foreman | Standard | 0.2 | 17 | 17 | 34.2 |
| | CrossLayer | 0.0 | 4.2 | 5.6 | 9.8 |
| Football | Standard | 0.1 | 20.4 | 16.6 | 37.1 |
| | CrossLayer | 0.0 | 5.2 | 4.0 | 9.2 |
| Stefan | Standard | 0.5 | 15 | 21.5 | 37.0 |
| | CrossLayer | 0.0 | 4.2 | 5.2 | 9.4 |
| Highway | Standard | 0.3 | 9.1 | 23.8 | 33.2 |
| | CrossLayer | 0.0 | 2.5 | 6.1 | 8.6 |
| Bus | Standard | 0.4 | 14 | 20.9 | 35.3 |
| | CrossLayer | 0.1 | 3.4 | 5.3 | 8.8 |
| Coast-gurad | Standard | 0.4 | 28.6 | 19.4 | 48.4 |
| | CrossLayer | 9.7 | 0.0 | 6.1 | 3.6 |
| Carphone | Standard | 2.4 | 6.0 | 8.9 | 17.3 |
| | CrossLayer | 0.0 | 1.9 | 3.2 | 5.1 |
| Mobile | Standard | 0.3 | 10.2 | 21.1 | 31.6 |
| | CrossLayer | 0.0 | 2.5 | 4.9 | 7.4 |
| Paris | Standard | 0.3 | 23 | 16.3 | 39.6 |
| | CrossLayer | 0.1 | 4.6 | 2.9 | 7.6 |
| Suzie | Standard | 0.0 | 12.8 | 10.3 | 23.1 |
| | CrossLayer | 0.0 | 2.3 | 1.8 | 4.1 |
| Akiyo | Standard | 0.2 | 13.6 | 16.2 | 30 |
| | CrossLayer | 0.0 | 3.3 | 3.1 | 6.4 |
| Bridge | Standard | 0.1 | 9.0 | 21.5 | 30.6 |
| | CrossLayer | 0.0 | 1.6 | 5.7 | 7.3 |
| Container | Standard | 0.5 | 28 | 26.4 | 54.9 |
| | CrossLayer | 0.2 | 10.5 | 3.3 | 14 |
| News | Standard | 0.1 | 19.1 | 23.4 | 42.6 |
| | CrossLayer | 0.1 | 5.0 | 6.2 | 11.3 |
| Hall | Standard | 0.2 | 5.9 | 23 | 29.1 |
| | CrossLayer | 0.0 | 1.7 | 5.2 | 6.9 |

Table 8.4: Detailed Analysis of % I, P and B Frame Loss for the Experiments Presented in Table 8.3

# 9 CONCLUSION AND FUTURE WORK

## 9.1 CONCLUSION

In this thesis we have investigated performance issues of real time multimedia data transfer over lossy wireless links over both single and multi-path network transport protocols. We investigated the performance of voice data (VoIP) over long delay satellite links with Datagram Congestion Control Protocol (DCCP). Our investigation lead us to believe that the performance of voice data over such long delay network links could significantly be improved by considering various techniques involving the DCCP's rate estimations and feedback acknowledgement methods. We have proposed both mitigation and improvement techniques as part of this thesis for DCCP to improve performance for VoIP data over long delay wireless network links.

During our investigation into DCCP as a focus for single-path transport protocol for real-time and multimedia streaming data delivery, we identified packet reordering as a critical performance degrading issue. As unreliable transport protocols such as DCCP by design do not maintain any retransmission strategy for out of order data packets and out of order packets are silently discarded by the transport protocol, DCCP's performance for real-time data suffers significantly in presence of packet reordering. To mitigate this issue in this single-path unreliable transport layer protocol, we have proposed a dynamic buffer based solution that can significantly improve the performance of DCCP without introducing a high level of jitter in real-time data delivery.

Packet reordering, and as a consequence, receiver's buffer blocking is equally a serious problem for multi-path transport protocols such as CMT-SCTP. Out of order packets arriving over dissimilar network paths cause receiver's buffer to become filled up quickly and eventually the sender ceases to transmit any new data causing a buffer block at the receiver. CMT-SCTP's inherent lack of capability to cope with asymmetric network paths prevents it from aggregating the full transmission capacity, that is spread over multiple paths. To overcome this issue, we have proposed a novel delay aware packet scheduling algorithm, which significantly improves CMT-SCTP bandwidth aggregation ability.

In order to improve CMT-SCTP's performance in the lossy network scenario, we have shown that by integrating erasure coding within the transport protocol's mechanisms, CMT-SCTP's multi-path transmission performance can significantly be improved. Since the impact of lossy network has the same impact on the receiver's buffer as packet reordering does, this network coding based proposal is equally useful for minimizing receiver's buffer blocking problem and improving overall in-time packet delivery, therefore ensuring better performance for real-time multimedia data transfer.

## 9.2 FUTURE WORK

Our contributions presented in this thesis could be augmented and extended in several ways:

- First, although DCCP was standardized by IETF around 2006 [13] and both Linux and FreeBSD has DCCP network stack implemented in the kernel, yet the number of multimedia applications using DCCP, particularly for VoIP, is very limited. During our research with VoIP over DCCP, we faced the challenge of encapsulating different voice codec encoded data into DCCP data packets for end-to-end transmission. It would be interesting to evaluate and quantify DCCP's performance with our proposal in more dynamic network scenarios with real telephony applications such as Asterisk [141, 189].

- Our contributions in Chapter 7 could be extended by experimenting and integrating with various other error correcting codes such as Fountain Codes which have already proven to be beneficial for various TCP variants under lossy network scenarios. A proper analytical modeling of CMT-SCTP with different network coding schemes would also be immensely useful for the research community. CMT-SCTP as a multi-path protocol could potentially consider pluggable error correction coding and congestion control within the transport layer, similar to DCCP's pluggable congestion algorithm architecture.

- Delay Aware Packet Scheduling (DAPS) proposed in Chapter 6 presents an interesting efficient solution for bandwidth aggregating data transmission protocols or even distributed data transfer systems such as in BitTorrent or Named Data Networking (NDN) [190]. Although we validated DAPS analytically in our work and presented simulation results to verify our proposal for asymmetric multi-path scenarios, what remains to be seen is how DAPS performs in real-world implementation inside the transport protocols and applications under changing network conditions. Varying network characteristics within an ongoing DAPS data transfer session is likely to introduce more difficulty in managing the DAPS parameters such as delay, buffered packets at the sender, packet sequencing etc. We believe more research and investigation is necessary to mitigate and address all issues to make DAPS more versatile under a broad range of network environments. Also

WebRTC, already using SCTP as a transport standard, opens up new challenges, where DAPS could be both relevant and beneficial.

# BIBLIOGRAPHY

[1] Steve Jobs, "Commencement address to Stanford graduates," 2005. [Online]. Available: http://www.youtube.com/watch?v=UF8uR6Z6KLc

[2] A. Sathiaseelan and G. Fairhurst, "Tcp-friendly rate control (tfrc) for bursty media flows." *Computer Communications*, vol. 34, no. 15, pp. 1836–1847, 2011. [Online]. Available: http://dblp.uni-trier.de/db/journals/comcom/comcom34.html#SathiaseelanF11

[3] "Cisco Visual Networking Index: Forecast and Methodology, 2012-2017." [Online]. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html

[4] "How Much Traffic Does Streaming Video Streaming Generate?" [Online]. Available: http://www.networkworld.com/community/blog/how-much-traffic-does-streaming-video-streaming-generate

[5] "Online Video is Causing Seismic Shifts in Internet Traffic." [Online]. Available: http://www.streamingmedia.com/Articles/Editorial/Featured-Articles/Online-Video-is-Causing-Seismic-Shifts-in-Internet-Traffic-84156.aspx

[6] "Goolge Voice." [Online]. Available: https://www.google.com/voice/

[7] "Google Voice, Wikipedia." [Online]. Available: http://en.wikipedia.org/wiki/Google_voice

[8] "Viber." [Online]. Available: http://www.viber.com

[9] "Viber, Wikipedia." [Online]. Available: http://en.wikipedia.org/wiki/Viber

[10] "Vine." [Online]. Available: https://vine.co

[11] "Skype Video Messaging." [Online]. Available: http://www.skype.com/en/features/video-messaging/

[12] "Mobile Data Traffic To Grow 300% Globally By 2017 Led By Video, Web Use." [Online]. Available: http://techcrunch.com/2013/07/03/mobile-data-use-to-grow-300-globally-by-2017

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

[14] J. R. Iyengar, P. D. Amer, and R. Stewart, "Concurrent multipath transfer using sctp multihoming over independent end-to-end paths," *IEEE/ACM Trans. Netw.*, vol. 14, no. 5, pp. 951–964, Oct. 2006. [Online]. Available: http://dx.doi.org/10.1109/TNET.2006.882843

[15] "SIGTRAN." [Online]. Available: http://www.ss7-training.net

[16] "SIGTRAN." [Online]. Available: http://www.sigtran.ss7box.com

[17] "IETF SIGTRAN Working Group." [Online]. Available: http://datatracker.ietf.org/wg/sigtran/charter

[18] "Signalling System 7." [Online]. Available: http://www.telecomspace.com/ss7.html

[19] "WebRTC." [Online]. Available: http://www.w3.org/TR/webrtc/

[20] IETF, "Hypertext Transfer Protocol (HTTP) 2.0," 2012. [Online]. Available: https://datatracker.ietf.org/wg/httpbis/charter

[21] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1," RFC 2068 (Proposed Standard), Internet Engineering Task Force, Jan. 1997, obsoleted by RFC 2616. [Online]. Available: http://www.ietf.org/rfc/rfc2068.txt

[22] E. Rescorla and N. Modadugu, "Datagram Transport Layer Security Version 1.2," RFC 6347 (Proposed Standard), Internet Engineering Task Force, Jan. 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6347.txt

[23] R. Stewart, "Stream Control Transmission Protocol," RFC 4960 (Proposed Standard), Internet Engineering Task Force, Sep. 2007, updated by RFCs 6096, 6335. [Online]. Available: http://www.ietf.org/rfc/rfc4960.txt

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

[25] S. Floyd and E. Kohler, "Profile for DCCP Congestion Control ID 4: TCP-Friendly Rate Control for Small Packets," IETF, Draft, Jun. 2006.

[26] 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.

[27] "IPSTAR Broadband Satellite Operatior." [Online]. Available: http://ipstar.com

[28] "Inmarsat." [Online]. Available: http://www.inmarsat.com

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

[30] H. Adhari, T. Dreibholz, M. Becke, E. P. Rathgeb, and M. Tüxen, "Evaluation of concurrent multipath transfer over dissimilar paths," in *Proceedings of the 2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications*, ser. WAINA '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 708–714. [Online]. Available: http://dx.doi.org/10.1109/WAINA.2011.92

[31] G. Sarwar, R. Boreli, E. Lochin, and A. Mifdaoui, "Performance evaluation of multipath transport protocol in heterogeneous network environments," in *Communications and Information Technologies (ISCIT), 2012 International Symposium on*, 2012, pp. 985–990.

[32] G. Sarwar, P. U. Tournoux, R. Boreli, and E. Lochin, "ecmt-sctp: Improving performance of multipath sctp with erasure coding over lossy links," in *IEEE Conference on Local Computer Networks (LCN)*. Sydney Australia: IEEE, October 2013, p. 8.

[33] G. Sarwar, R. Boreli, E. Lochin, A. Mifdaoui, and G. Smith, "Mitigating receiver's buffer blocking by delay aware packet scheduling in multipath data transfer," in *The 3rd International Workshop on Protocols and Applications with Multi-Homing Support*, Barcelona, Spain, December 2012, p. 6.

[34] G. Sarwar, R. Boreli, E. Lochin, and A. Mifdaour, "Performance evaluation of multipath transport protocol in asymmetric heterogeneous network environment," in *International Symposium on Communications and Information Technologies (ISCIT)*, Gold Coast, Australia, October 2012, p. 6.

[35] G. Sarwar, R. Boreli, and E. Lochin, "On the quality of voip with dccp for satellite communications," *International Journal of Satellite Communications and Networking*, vol. 30, no. Issue 4, pp. 163–180, February 2012.

[36] ——, "Mitigating the impact of packet reordering to maximize performance of multimedia applications," in *IEEE International Conference on Communications (ICC)*, Japan, June 2011, p. 6.

[37] ——, "Xstream-x264: A tool for real-time h.264 encoding and streaming with cross-layer integration," in *International Conference on Multimedia and Expo (ICME 2011)*, Barcelona, Spain, July 2011, p. 4.

[38] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550 (INTERNET STANDARD), Internet Engineering Task Force, Jul. 2003, updated by RFCs 5506, 5761, 6051, 6222. [Online]. Available: http://www.ietf.org/rfc/rfc3550.txt

[39] T. Friedman, R. Caceres, and A. Clark, "RTP Control Protocol Extended Reports (RTCP XR)," RFC 3611 (Proposed Standard), Internet Engineering Task Force, Nov. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3611.txt

[40] H. Schulzrinne, A. Rao, and R. Lanphier, "Real Time Streaming Protocol (RTSP)," RFC 2326 (Proposed Standard), Internet Engineering Task Force, Apr. 1998. [Online]. Available: http://www.ietf.org/rfc/rfc2326.txt

[41] D. D. Clark and D. L. Tennenhouse, "Architectural considerations for a new generation of protocols," in *Proceedings of the ACM Symposium on Communications Architectures &Amp; Protocols*, ser. SIGCOMM '90. New York, NY, USA: ACM, 1990, pp. 200–208. [Online]. Available: http://doi.acm.org/10.1145/99508.99553

[42] Talaat, Mohammad A., Magdi A. Koutb, and Hoda S. Sorour, "A survey on unicast congestion control protocols for media traffic," *International Journal Computer Science and Network Security*, vol. 9.3, pp. 254–261, 2009.

[43] D. Sisalem and A. Wolisz, "LDA+ TCP-Friendly Adaptation: A Measurement and Comparison Study," in *in the 10th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'2000*, 2000, pp. 25–28.

[44] N. Aboobaker, D. Chanady, M. Gerla, and M. Y. Sanadidi, "Streaming media congestion control using bandwidth estimation," in *Proceedings of the 5th IFIP/IEEE International Conference on Management of Multimedia Networks and Services: Management of Multimedia on the Internet*, ser. MMNS '02. London, UK, UK: Springer-Verlag, 2002, pp. 89–100. [Online]. Available: http://dl.acm.org/citation.cfm?id=645750.667719

[45] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, "Tcp westwood: Bandwidth estimation for enhanced transport over wireless links," in *Proceedings of the 7th annual international conference on Mobile computing and networking*, ser. MobiCom '01. New York, NY, USA: ACM, 2001, pp. 287–297. [Online]. Available: http://doi.acm.org/10.1145/381677.381704

[46] J. Yan, K. Katrinis, M. May, and B. Plattner, "Media- and tcp-friendly congestion control for scalable video streams," *Multimedia, IEEE Transactions on*, vol. 8, no. 2, pp. 196–206, 2006.

[47] C. Bouras, A. Gkamas, V. Kapoulas, V. Papapanagiotou, K. Stamos, and G. Zaoudis, "Video transmission over tfrc using cross-layer power management," in *Proceedings of the 17th international conference on Software, Telecommunications and Computer Networks*, ser. SoftCOM'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 333–337. [Online]. Available: http://dl.acm.org/citation.cfm?id=1719770.1719837

[48] R. Rejaie, M. Handley, and D. Estrin, "Rap: An end-to-end rate-based congestion control mechanism for realtime streams in the internet," in *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, 1999, pp. 1337–1345 vol.3.

[49] P. Papadimitriou and V. Tsaoussidis, "Ssvp: A congestion control scheme for real-time video streaming," *Comput. Netw.*, vol. 51, no. 15, pp. 4377–4395, Oct. 2007. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2007.06.018

[50] S. H. Choi and M. Handley, "Fairer tcp-friendly congestion control protocol for multimedia streaming applications," in *Proceedings of the 2007 ACM CoNEXT conference*, ser. CoNEXT '07. New York, NY, USA: ACM, 2007, pp. 54:1–54:2. [Online]. Available: http://doi.acm.org/10.1145/1364654.1364717

[51] P. Papadimitriou and V. Tsaoussidis, "A rate control scheme for adaptive video streaming over the internet," in *Communications, 2007. ICC '07. IEEE International Conference on*, 2007, pp. 616–621.

[52] J. Postel, "User Datagram Protocol," RFC 768 (INTERNET STANDARD), Internet Engineering Task Force, Aug. 1980. [Online]. Available: http://www.ietf.org/rfc/rfc768.txt

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

[54] S. Floyd and E. Kohler, "Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 2: TCP-like Congestion Control," RFC 4341 (Proposed Standard), Internet Engineering Task Force, Mar. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4341.txt

[55] ——, "Profile for Datagram Congestion Control Protocol (DCCP) Congestion ID 4: TCP-Friendly Rate Control for Small Packets

(TFRC-SP)," RFC 5622 (Experimental), Internet Engineering Task Force, Aug. 2009, updated by RFC 6323. [Online]. Available: http://www.ietf.org/rfc/rfc5622.txt

[56] ——, "TCP Friendly Rate Control (TFRC): The Small-Packet (SP) Variant," RFC 4828 (Experimental), Internet Engineering Task Force, Apr. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4828.txt

[57] J. C. R. Bennett, C. Partridge, and N. Shectman, "Packet reordering is not pathological network behavior," *IEEE/ACM Trans. Netw.*, vol. 7, no. 6, pp. 789–798, Dec. 1999. [Online]. Available: http://dx.doi.org/10.1109/90.811445

[58] M. Przybylski, B. Belter, and A. Binczewski, "Shall we worry about packet reordering," in *Computational Methods in Science and Technology*, 2005, pp. 141–146.

[59] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, and D. Towsley, "Measurement and classification of out-of-sequence packets in a tier-1 ip backbone," *IEEE/ACM Trans. Netw.*, vol. 15, no. 1, pp. 54–66, Feb. 2007. [Online]. Available: http://dx.doi.org/10.1109/TNET.2006.890117

[60] L. Gharai, C. Perkins, and T. Lehman, "Packet reordering, high speed networks and transport protocol performance," in *Computer Communications and Networks, 2004. ICCCN 2004. Proceedings. 13th International Conference on*, 2004, pp. 73–78.

[61] T. I. E. T. F. (IETF), "Mptcp working group." [Online]. Available: http://datatracker.ietf.org/wg/mptcp/charter/

[62] I. van Beijnum, "Multipath TCP," *IETF Journal*, vol. 5, September 2009.

[63] M. Handley, "Multipath TCP: Goals and Background," Presentation at Tsinghua University and Bristol University, November 2009. [Online]. Available: http://nrg.cs.ucl.ac.uk/mptcp/presentations/mptcp-bof-print.pdf

[64] A. Ford, C. Raiciu, M. Handley, "TCP Extensions for Multipath Operation with Multiple Addresses," IETF Draft, March 2010. [Online]. Available: http://tools.ietf.org/html/draft-ietf-mptcp-multiaddressed-12

[65] O.-e. m. T. I. van Beijnum, "One-ended Multipath TCP," The Internet Engineering Task Force (IETF), Internet Draft, May 2009. [Online]. Available: http://tools.ietf.org/html/draft-van-beijnum-1e-mp-tcp-00

[66] C. Raiciu, M. Handley, D. Wischik, "Coupled Multipath-Aware Congestion Control," IETF Draft, July 2011. [Online]. Available: http://tools.ietf.org/html/draft-ietf-mptcp-congestion-07

[67] H. Han, S. Shakkottai, C. V. Hollot, R. Srikant, and D. Towsley, "Multi-path tcp: a joint congestion control and routing scheme to exploit path diversity in the internet," *IEEE/ACM Trans. Netw.*, vol. 14, no. 6, pp. 1260–1271, Dec. 2006. [Online]. Available: http://dx.doi.org/10.1109/TNET.2006.886738

[68] L. Magalhaes and R. Kravets, "Transport level mechanisms for bandwidth aggregation on mobile hosts," in *Network Protocols, 2001. Ninth International Conference on*. IEEE, 2001, pp. 165–171.

[69] H.-Y. Hsieh, K.-H. Kim, Y. Zhu, and R. Sivakumar, "A receiver-centric transport protocol for mobile hosts with heterogeneous wireless interfaces," in *Proceedings of the 9th annual international conference on Mobile computing and networking*, ser. MobiCom '03. New York, NY, USA: ACM, 2003, pp. 1–15. [Online]. Available: http://doi.acm.org/10.1145/938985.938987

[70] M. Welzl, D. Damjanovic, S. Gjessing, "MulTFRC: TFRC with weighted fairness," IETF Draft, July 2010. [Online]. Available: http://tools.ietf.org/html/draft-irtf-iccrg-multfrc-01

[71] D. Damjanovic, "PhD Thesis."

[72] Michael Welzl, "Webpage." [Online]. Available: http://heim.ifi.uio.no/michawe/

[73] J. Crowcroft and P. Oechslin, "Differentiated end-to-end internet services using a weighted proportional fair sharing tcp," *SIGCOMM Comput. Commun. Rev.*, vol. 28, no. 3, pp. 53–69, Jul. 1998. [Online]. Available: http://doi.acm.org/10.1145/293927.293930

[74] "Multi-TFRC Working Group." [Online]. Available: http://tools.ietf.org/group/irtf/trac/wiki/ICCRG_multfrc

[75] H. Y. Hsieh and R. Sivakumar, "pTCP: An End-to-End transport layer protocol for striped connections," in *ICNP '02: Proceedings of the 10th IEEE International Conference on Network Protocols*. Washington, DC, USA: IEEE Computer Society, 2002, pp. 24–33. [Online]. Available: http://portal.acm.org/citation.cfm?id=645532.656327

[76] H.-Y. Hsieh and R. Sivakumar, "A transport layer approach for achieving aggregate bandwidths on multi-homed mobile hosts," in *Proceedings of the 8th annual international conference on Mobile computing and networking*, ser. MobiCom '02. New

York, NY, USA: ACM, 2002, pp. 83–94. [Online]. Available: http://doi.acm.org/10.1145/570645.570656

[77] C.-H. Chang and T.-Y. Chung, "A cross-layer design to achieve stable and efficient parallel tcp over next generation wireless networks." *Ad Hoc & Sensor Wireless Networks*, vol. 8, no. 3-4, pp. 183–210, 2009. [Online]. Available: http://dblp.uni-trier.de/db/journals/ahswn/ahswn8.html#ChangC09

[78] T. Ito, H. Ohsaki, and M. Imase, "Gridftp-apt: Automatic parallelism tuning mechanism for data transfer protocol gridftp," in *Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid*, ser. CCGRID '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 454–461. [Online]. Available: http://dx.doi.org/10.1109/CCGRID.2006.51

[79] H.-H. Wan and Y.-K. Kwok, "High data rate video transmission using parallel tcp connections: approaches and performance evaluation," *J. Supercomput.*, vol. 35, no. 2, pp. 119–139, Feb. 2006. [Online]. Available: http://dx.doi.org/10.1007/s11227-006-3071-8

[80] M. Chen and A. Zakhor, "Multiple tfrc connections based rate control for wireless networks," *Trans. Multi.*, vol. 8, no. 5, pp. 1045–1062, Oct. 2006. [Online]. Available: http://dx.doi.org/10.1109/TMM.2006.879837

[81] K. Chebrolu, B. Raman, and R. R. Rao, "A network layer approach to enable tcp over multiple interfaces," *Wirel. Netw.*, vol. 11, no. 5, pp. 637–650, Sep. 2005. [Online]. Available: http://dx.doi.org/10.1007/s11276-005-3518-5

[82] I. A. Valdovinos and J. A. P. Díaz, "Tcp extension to send traffic simultaneously through multiple heterogeneous network interfaces," in *Proceedings of the 2009 Mexican International Conference on Computer Science*, ser. ENC '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 89–94. [Online]. Available: http://dx.doi.org/10.1109/ENC.2009.28

[83] V. Sharma, V. Subramanian, K. Kar, S. Kalyanaraman, K. Ramakrishnan, and B. Ganguly, "A multi-path transport protocol to exploit network diversity in airborne networks," in *Military Communications Conference, 2008. MILCOM 2008. IEEE*, 2008, pp. 1–7.

[84] A. Argyriou and V. Madisetti, "Bandwidth aggregation with sctp," in *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, vol. 7, 2003, pp. 3716–3721 vol.7.

[85] A. A. E. Al, T. Saadawi, and M. Lee, "A transport layer load sharing mechanism for mobile wireless hosts," in *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, ser. PERCOMW '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 87–. [Online]. Available: http://dl.acm.org/citation.cfm?id=977405.978603

[86] Y. Hasegawa, I. Yamaguchi, Takayuki Hama, H. Shimonishi and T. Murase, "Improved data distribution for multipath tcp communication," in *Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE*, vol. 1, 2005, pp. 5 pp.–.

[87] D. Sarkar, "A concurrent multipath tcp and its markov model," in *Communications, 2006. ICC '06. IEEE International Conference on*, vol. 2, 2006, pp. 615–620.

[88] J. Anand and D. Sarkar, "cmprtcp: Concurrent multi-path real-time tcp," in *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, 2007, pp. 2031–2036.

[89] R. Stewart, M. Ramalho, Q. Xie, M. Tuexen, and P. Conrad, "Stream Control Transmission Protocol (SCTP) Partial Reliability Extension," RFC 3758 (Proposed Standard), Internet Engineering Task Force, May 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3758.txt

[90] M. F. Tsai, N. Chilamkurti, J. Park, and C. K. Shieh, "Multi-path transmission control scheme combining bandwidth aggregation and packet scheduling for real-time streaming in multi-path environment," *Communications, IET*, vol. 4, no. 8, pp. 937–945, 2010.

[91] M. Honda, Y. Nishida, L. Eggert, P. Sarolahti, and H. Tokuda, "Multipath congestion control for shared bottleneck," in *International Workshop on Protocols for Future, Large-Scale and Diverse Network Transports (PFLDNeT)*, Akihabara, Tokyo, Japan, 2009.

[92] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, "Stream Control Transmission Protocol," RFC 2960 (Proposed Standard), Internet Engineering Task Force, Oct. 2000, obsoleted by RFC 4960, updated by RFC 3309. [Online]. Available: http://www.ietf.org/rfc/rfc2960.txt

[93] L. Ong and J. Yoakum, "An Introduction to the Stream Control Transmission Protocol (SCTP)," RFC 3286 (Informational), Internet Engineering Task Force, May 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3286.txt

[94] R. Stewart, M. Ramalho, Q. Xie, M. Tuexen, and P. Conrad, "Stream control transmission protocol (sctp) partial reliability extension," United States, 2004.

[95] R. Braden, D. Borman, and C. Partridge, "Computing the Internet checksum," RFC 1071, Internet Engineering Task Force, Sep. 1988, updated by RFC 1141. [Online]. Available: http://www.ietf.org/rfc/rfc1071.txt

[96] CERT Advisory, "SYN Flooding," September 1996. [Online]. Available: http://www.cert.org/advisories/CA-1996-21.html

[97] T. Dreibholz, R. Seggelmann, M. Tüxen, and E. P. Rathgeb, "Transmission Scheduling Optimizations for Concurrent Multipath Transfer," in *Proceedings of the 8th International Workshop on Protocols for Future, Large-Scale and Diverse Network Transports (PFLDNeT)*, vol. 8, Lancaster, Pennsylvania/U.S.A., Nov. 2010. [Online]. Available: http://www.tdr.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/SCTP/Paper/PFLDNeT2010.pdf

[98] J. Prokkola, P. H. J. Perälä, M. Hanski, and E. Piri, "3g/hspa performance in live networks from the end user perspective," in *Proceedings of the 2009 IEEE international conference on Communications*, ser. ICC'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 1152–1157. [Online]. Available: http://dl.acm.org/citation.cfm?id=1817271.1817485

[99] C. Raiciu, C. Paasch, S. Barre, A. Ford, M. Honda, F. Duchene, O. Bonaventure, and M. Handley, "How hard can it be? designing and implementing a deployable multipath tcp," in *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, ser. NSDI'12. Berkeley, CA, USA: USENIX Association, 2012, pp. 29–29. [Online]. Available: http://dl.acm.org/citation.cfm?id=2228298.2228338

[100] J. Iyengar, P. Amer, and R. Stewart, "Retransmission policies for concurrent multipath transfer using sctp multihoming," in *Networks, 2004. (ICON 2004). Proceedings. 12th IEEE International Conference on*, vol. 2, 2004, pp. 713–719 vol.2.

[101] J. Liu, H. Zou, J. Dou, and Y. Gao, "Rethinking retransmission policy in concurrent multipath transfer," in *Intelligent Information Hiding and Multimedia Signal Processing, 2008. IIHMSP '08 International Conference on*, 2008, pp. 1005–1008.

[102] Y. Qiao, E. Fallon, J. Murphy, L. Murphy, and A. Hanley, "Path selection of sctp fast retransmission in multi-homed wireless environments," in *MWCN/PWC*, 2008, pp. 447–458.

[103] M. C. Chan and R. Ramjee, “Tcp/ip performance over 3g wireless links with rate and delay variation,” in *Proceedings of the 8th annual international conference on Mobile computing and networking*, ser. MobiCom ’02. New York, NY, USA: ACM, 2002, pp. 71–82. [Online]. Available: http://doi.acm.org/10.1145/570645.570655

[104] S. Boussen, N. Tabbane, and S. Tabbane, “Performance analysis of sctp protocol in wifi network,” in *Proceedings of the 2009 Fourth International Conference on Computer Sciences and Convergence Information Technology*, ser. ICCIT ’09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 178–182. [Online]. Available: http://dx.doi.org/10.1109/ICCIT.2009.30

[105] X. D. R. Stewart, M. Tuexen, “ECN for Stream Control Transmission Protocol,” IETF Internet Draft, February 2012. [Online]. Available: http://tools.ietf.org/html/draft-stewart-tsvwg-sctpecn-03

[106] D. Tian and G. AlRegib, “Parity streams: a novel fec scheme with the stream control transmission protocol,” *Communications Letters, IEEE*, vol. 10, no. 6, pp. 498–500, 2006.

[107] Y. Cui, X. Wang, H. Wang, G. Pan, and Y. Wang, “Fmtcp: A fountain code-based multipath transmission control protocol,” in *Proceedings of the 2012 IEEE 32nd International Conference on Distributed Computing Systems*, ser. ICDCS ’12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 366–375. [Online]. Available: http://dx.doi.org/10.1109/ICDCS.2012.23

[108] D. J. C. MacKay, “Fountain codes,” *Communications, IEE Proceedings-*, vol. 152, no. 6, pp. 1062–1068, 2005.

[109] Y. Hwang, B. O. Obele, and H. Lim, “Multipath transport protocol for heterogeneous multi-homing networks,” in *Proceedings of the ACM CoNEXT Student Workshop*, ser. CoNEXT ’10 Student Workshop. New York, NY, USA: ACM, 2010, pp. 5:1–5:2. [Online]. Available: http://doi.acm.org/10.1145/1921206.1921212

[110] K. Lam, J. Chapin, and V. W. S. Chan, “Performance analysis and optimization of multipath tcp,” in *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*, 2011, pp. 695–700.

[111] S. Benayoune, N. Achir, K. Boussetta, and K. Chen, “A mac centric cross layer approach for h. 264 video streaming over hsdpa,” *Journal of Communications*, vol. 4, no. 9, pp. 691–699, 2009.

[112] “H.264/AVC JM Reference Software.” [Online]. Available: http://iphome.hhi.de/suehring/tml/

[113] S. Mccanne, S. Floyd, and K. Fall, "ns2 (network simulator 2)," http://www-nrg.ee.lbl.gov/ns/. [Online]. Available: http://www-nrg.ee.lbl.gov/ns

[114] R. Soni, N. Chilamkurti, G. Giambene, and S. Zeadally, "A cross-layer design for h.264 video stream over wireless local area networks," in *Proceedings of the International Symposium on Computer Science and its Applications*, ser. CSA '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 387–392. [Online]. Available: http://dx.doi.org/10.1109/CSA.2008.40

[115] Z. Liu, H. Liu, and Y. Wang, "Cross layer adaptation for h.264 video multicasting over wireless lan," in *Multimedia and Expo, 2006 IEEE International Conference on*, 2006, pp. 1121–1124.

[116] B. J. Oh and C. W. Chen, "A cross-layer adaptation hcca mac for qos-aware h.264 video communications over wireless mesh networks," in *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, 2010, pp. 2259–2262.

[117] C. Atici and M. Sunay, "Improving the performance of wireless h.264 video broadcasting through a cross-layer design," in *Broadband Multimedia Systems and Broadcasting, 2009. BMSB '09. IEEE International Symposium on*, 2009, pp. 1–6.

[118] "EvalVid-RA - The Rate Adaptive Evalvid tool-set." [Online]. Available: http://www.item.ntnu.no/~arnelie/Evalvid-RA.htm

[119] K. J. Lie A, "Evalvid-ra: trace driven simulation of rate adaptive mpeg-4 vbr video," 2008.

[120] "EvalVid - A Video Quality Evaluation Tool-set." [Online]. Available: http://www.tkn.tu-berlin.de/research/evalvid/

[121] "MPEG4IP." [Online]. Available: http://www.mpeg4ip.net/

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

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

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

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

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

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

[128] C. Perkins, "RTP and the Datagram Congestion Control Protocol (DCCP)," RFC 5762 (Proposed Standard), Internet Engineering Task Force, Apr. 2010, updated by RFC 6773. [Online]. Available: http://www.ietf.org/rfc/rfc5762.txt

[129] T-REC-P.800-199608-I, "P.800 : Methods for subjective determination of transmission quality." [Online]. Available: http://www.itu.int/rec/T-REC-P.800-199608-I/en

[130] 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.

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

[132] ITU-T Recommendation P.862, "P.862: Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," 2001.

[133] ITU-T Recommendation P.862.1, "P.862.1 : Mapping function for transforming P.862 raw result scores to MOS-LQO ," 2003.

[134] A. Meddahi and H. Afifi, "Packet-E-Model: e-model for VoIP quality evaluation," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 50, no. 15, pp. 2659–2675, Oct. 2006.

[135] M. Handley, S. Floyd, J. Padhye, and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification," RFC 3448 (Proposed Standard), Internet Engineering Task Force, Jan. 2003, obsoleted by RFC 5348. [Online]. Available: http://www.ietf.org/rfc/rfc3448.txt

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

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

[138] ——, "Quick-Start for the Datagram Congestion Control Protocol (DCCP)," RFC 5634 (Experimental), Internet Engineering Task Force, Aug. 2009. [Online]. Available: http://www.ietf.org/rfc/rfc5634.txt

[139] S. Floyd and E. Kohler, "Faster Restart for TCP Friendly Rate Control (TFRC)," IETF, Draft, 2006.

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

[141] "Asterisk PBX," http://www.asterisk.org/. [Online]. Available: http://www.asterisk.org

[142] 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.

[143] 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.

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

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

[146] 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, pp. 19–24.

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

[148] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "TCP Vegas: new techniques for congestion detection and avoidance," pp. 24–35, 1994.

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

[150] 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 2009.

[151] Speex codec. [Online]. Available: http://www.speex.org

[152] 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.

[153] G. Fairhurst and A. Sathiaseelan, "Updating TCP to support Rate-Limited Traffic," TCPM Working Group, IETF Internet Draft, February 2013. [Online]. Available: http://tools.ietf.org/html/draft-ietf-tcpm-newcwv-00

[154] N. Dukkipati, T. Refice, Y. Cheng, J. Chu, T. Herbert, A. Agarwal, A. Jain, and N. Sutin, "An argument for increasing tcp's initial congestion window," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 3, pp. 26–33, Jun. 2010. [Online]. Available: http://doi.acm.org/10.1145/1823844.1823848

[155] T. Phelan, G. Fairhurst, and C. Perkins, "DCCP-UDP: A Datagram Congestion Control Protocol UDP Encapsulation for NAT Traversal," RFC 6773 (Proposed Standard), Internet Engineering Task Force, Nov. 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6773.txt

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

[157] S. Wenger, "H.264/avc over ip," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 645–656, 2003.

[158] "TCP stress testing script for NS-2." [Online]. Available: http://www.hamilton.ie/net/tcptesting.zip

[159] L. Andrew, C. Marcondes, S. Floyd, L. Dunn, and L. Eggert, "Towards a common tcp evaluation suite," in *Sixth International Workshop on Protocols for FAST Long-Distance Networks*, 2008. [Online]. Available: https://fit.nokia.com/lars/papers/2008-pfldnet.pdf

[160] ITU-T, "G.1070: Opinion model for video-telephony applications," 2007.

[161] Tom Phelan, "DCCP-TP - DCCP Implementation Optimized for Portability." [Online]. Available: http://www.phelan-4.com/dccp-tp/tiki-index.php

[162] T. Dreibholz, M. Becke, E. Rathgeb, and M. Tuxen, "On the use of concurrent multipath transfer over asymmetric paths," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, 2010, pp. 1–6.

[163] P. Natarajan, N. Ekiz, E. Yilmaz, P. D. Amer, and J. Iyengar, "Non-renegable selective acknowledgments (nr-sacks) for sctp." IEEE International Conference on Network Protocols, October 2008.

[164] E. Yilmaz, N. Ekiz, P. Natarajan, P. D. Amer, J. T. Leighton, F. Baker, and R. R. Stewart, "Throughput analysis of non-renegable selective acknowledgments (nr-sacks) for sctp," *Comput. Commun.*, vol. 33, no. 16, pp. 1982–1991, October 2010. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2010.06.028

[165] K. Y. P. Sarolahti, M. Kojo and M. Hata, "Forward rto-recovery (f-rto): An algorithm for detecting spurious retransmission timeouts with tcp," The Internet Engineering Task Force (IETF), RFC 5682, September 2009. [Online]. Available: http://tools.ietf.org/html/rfc5682

[166] M. A. V. Paxson, "Computing tcp's retransmission timer," The Internet Engineering Task Force (IETF), RFC 2988, November 2000. [Online]. Available: http://tools.ietf.org/html/rfc2988

[167] "GNU Octave." [Online]. Available: http://www.gnu.org/software/octave/

[168] Z. Yi, *Analytical Modeling of SCTP and Load Balancing SCTP*. City University of New York, 2005. (Graduate Faculty in Engineering), 2005. [Online]. Available: http://books.google.com/books?id=xZd8NwAACAAJ

[169] J. K. Sundararajan, D. Shah, M. Médard, M. Mitzenmacher, and J. Barros, "Network coding meets tcp," in *INFOCOM 2009, IEEE*. IEEE, 2009, pp. 280–288.

[170] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[171] P. Tournoux, E. Lochin, J. Lacan, A. Bouabdallah, and V. Roca, "On-the-fly erasure coding for real-time video applications," *Multimedia, IEEE Transactions on*, vol. 13, no. 4, pp. 797–812, 2011.

[172] C. S. Q. Xie, R. R. Stewart and I. Rytina, "Sctp unreliable data mode extension," *IETF Draft*, 2001.

[173] P.-U. Tournoux, E. Lochin, H. Petander, and J. Lacan, "A packet error recovery scheme for vertical handovers mobility management protocols," in *MobiQuitous*, 2010, pp. 249–260.

[174] I. T. Union, "ITU-T recommendation G.114 - one-way transmission time," May 2003.

[175] T. T. Thai, E. Lochin, and J. Lacan, "Online multipath convolutional coding for real-time transmission," in *Packet Video Workshop (PV), 2012 19th International*. IEEE, 2012, pp. 41–46.

[176] A. . International Telecommunication Union, "ITU-T recommendation G.1070 opinion model for video-telephony applications," April 2007.

[177] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h.264/avc video coding standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 560–576, 2003.

[178] A. Huszak and S. Imre, "Analysing gop structure and packet loss effects on error propagation in mpeg-4 video streams," in *Communications, Control and Signal Processing (ISCCSP), 2010 4th International Symposium on*, 2010, pp. 1–5.

[179] "x264." [Online]. Available: http://www.videolan.org

[180] "ITU-T Rec. J.247 (08/08) Objective perceptual multimedia video quality measurement in the presence of a full reference." [Online]. Available: http://www.itu.int/rec/T-REC-J.247/en

[181] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of psnr in image/video quality assessment," *Electronics Letters*, vol. 44, no. 13, pp. 800–801, 2008.

[182] "ImageMagick." [Online]. Available: http://www.imagemagick.org/

[183] S. K. Bandyopadhyay, Z. Wu, P. Pandit, and J. M. Boyce, "An error concealment scheme for entire frame losses for h.264/avc," in *Sarnoff Symposium, 2006 IEEE*, 2006, pp. 1–4.

[184] "Xiph. Foundation." [Online]. Available: http://media.xiph.org/

[185] "YUV Video Sequences by Arizona State University." [Online]. Available: http://trace.eas.asu.edu/yuv/

[186] "Python Programming Language." [Online]. Available: http://www.python.org

[187] "Scalable Vector Graphics." [Online]. Available: http://www.w3.org/Graphics/SVG

[188] "HTML5 Specification." [Online]. Available: http://www.w3.org/TR/html5/

[189] "Wikipedia - Asterisk PBX." [Online]. Available: http://en.wikipedia.org/wiki/Asterisk_(PBX)

[190] "Named Data Networking." [Online]. Available: http://named-data.net