UNIVERSITY OF HELSINKI
FACULTY OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE
PHD THESIS

ENABLING MULTIPATH AND MULTICAST DATA TRANSMISSION IN LEGACY AND FUTURE INTERNET

TATIANA POLISHCHUK

TATIANA POLISHCHUK | ENABLING MULTIPATH AND MULTICAST DATA TRANSMISSION IN LEGACY AND FUTURE INTERNET A-2013-7

Department of Computer Science
Series of Publications A
Report A-2013-7

Enabling Multipath and Multicast Data Transmission in Legacy and Future Internet

Tatiana Polishchuk

To be presented, with the permission of the Faculty of Science of the University of Helsinki, for public criticism in Auditorium XII, Main Building, University of Helsinki, on September 9, 2013 at 12 o'clock.

University of Helsinki
Finland

Supervisor
Professor Jussi Kangasharju, University of Helsinki, Finland

Instructor
Professor Andrei Gurtov, Helsinki Institute for Information Technology
HIIT, Aalto University, Finland

Pre-examiners
Dr. Sergey Gorinsky, Institute IMDEA Networks, Spain
Prof. Guillaume Urvoy-Keller, Laboratoire I3S, CNRS, France

Opponent
Dr. Bob Briscoe, BT Research, England

Custos
Professor Jussi Kangasharju, University of Helsinki, Finland

Contact information

Department of Computer Science
P.O. Box 68 (Gustaf Hällströmin katu 2b)
FI-00014 University of Helsinki
Finland

Email address: postmaster@cs.helsinki.fi
URL: http://www.cs.Helsinki.fi/
Telephone: +358 9 1911, telefax: +358 9 191 51120

ISSN 1238-8645
ISBN 978-952-10-9006-6 (paperback)
ISBN 978-952-10-9007-3 (PDF)
Computing Reviews (1998) Classification:
Helsinki 2013
Helsinki University Print

Enabling Multipath and Multicast Data Transmission in Legacy and Future Internet

Tatiana Polishchuk

Department of Computer Science
P.O. Box 68, FI-00014 University of Helsinki, Finland
tatiana.polishchuk@hiit.fi
http://www.hiit.fi/ tpolishc

PhD Thesis, Series of Publications A, Report A-2013-7
Helsinki, 2013, 127 pages
ISSN 1238-8645
ISBN 978-952-10-9006-6 (paperback)
ISBN 978-952-10-9007-3 (PDF)

Abstract

The quickly growing community of Internet users is requesting multiple applications and services. At the same time the structure of the network is changing. From the performance point of view, there is a tight interplay between the application and the network design. The network must be constructed to provide an adequate performance of the target application.

In this thesis we consider how to improve the quality of users' experience concentrating on two popular and resource-consuming applications: bulk data transfer and real-time video streaming. We share our view on the techniques which enable feasibility and deployability of the network functionality leading to unquestionable performance improvement for the corresponding applications.

Modern mobile devices, equipped with several network interfaces, as well as multihomed residential Internet hosts are capable of maintaining multiple simultaneous attachments to the network. We propose to enable simultaneous multipath data transmission in order to increase throughput and speed up such bandwidth-demanding applications as, for example, file download. We design an extension for Host Identity Protocol (mHIP), and propose a multipath data scheduling solution on a wedge layer between IP and transport, which effectively distributes packets from a TCP connection over available paths. We support our protocol with a congestion control scheme and prove its ability to compete in a friendly manner against the legacy network protocols. Moreover, applying game-theoretic analytical modelling we investigate how the multihomed HIP multipath-enabled hosts coexist in the shared network.

The number of real-time applications grows quickly. Efficient and reliable transport of multimedia content is a critical issue of today's IP network design. In this thesis we solve scalability issues of the multicast dissemination trees controlled by the hybrid error correction. We propose a scalable multicast architecture for potentially large overlay networks. Our techniques address suboptimality of the adaptive hybrid error correction (AHEC) scheme in the multicast scenarios. A hierarchical multi-stage multicast tree topology is constructed in order to improve the performance of AHEC and guarantee QoS for the multicast clients. We choose an evolutionary networking approach that has the potential to lower the required resources for multimedia applications by utilizing the error-correction domain separation paradigm in combination with selective insertion of the supplementary data from parallel networks, when the corresponding content is available.

Clearly both multipath data transmission and multicast content dissemination are the future Internet trends. We study multiple problems related to the deployment of these methods.

Computing Reviews (1998) Categories and Subject Descriptors:

General Terms:
Thesis, Multipath Data Scheduling, HIP, Game Theory, Fair Resource Sharing, Multicast Scalability, HEC, Redundancy Optimization

Additional Key Words and Phrases:
Future Internet, TCP-fairness, TCP-friendliness, Real-time Multimedia Applications

Acknowledgements

I'm happy to express my gratitude to the exceptional people without whom this dissertation would not have been possible.

I thank my supervisor Jussi Kangasharju for his guidance, support and positive attitude throughout my PhD study at the University of Helsinki.

I'm truly grateful to my instructor Andrei Gurtov for his continuous support, encouragement and patience, and for providing me with an inspiring and fun working environment in HIIT networking research group.

I wish to thank the pre-examiners Dr. Sergey Gorinsky and Prof. Guillaume Urvoy-Keller for their valuable effort in reading the manuscript and for their helpful comments.

The results and findings presented in this thesis are to a large extent an outcome of the joint work with many excellent researchers in and outside HIIT. I would like to thank all my co-authors and collaborators from the University of Saarland and from the Karelian Research Center of Russian Academy of Sciences.

I'm very grateful to my colleagues and staff in HIIT for providing me with the opportunity and friendly atmosphere to carry out this PhD research work, for keeping things running so smoothly. I would like to acknowledge my colleagues from the Networking group, who were always eager to help and share their research ideas. Especially I want to thank Dmitry Kuptsov for his effort in implementation of my multipath ideas in the mHIP prototype, and for his generous permission to use the results and some pictures in this thesis. I also thank Ilya Nikolaevsky for the significant time he devoted to experimentation with mHIP.

I would like to thank Marina Kurten from the CS Department at the University of Helsinki for language preview of the manuscript.

I'm extremely grateful to my parents, Antonina and Oleg, and my sister Svetlana, for their endless care and support, love and constant help.

I'm thankful to my mother-in-law Nina, for her valuable help and encouragement throughout the process of writing my dissertation.

I would like to thank my dearest husband, Valentin, for supporting

and encouraging me through the long path of my study and research. His personal example and close attention to my work helped me to proceed on this thorny path. His critical feedback and advice had an essential impact on the results of my work.

And finally, the completion of this work would not be possible without the empathy of my loveliest daughters Snezhana and Diana. Thank you for your endless patience, true love and your place in my heart.

Contents

1 Introduction **3**

1.1 Research Problems and Scope 5

1.2 Research History 7

1.3 Contributions 8

1.4 Structure of the Thesis 8

2 Literature Review and Background **9**

2.1 Multipath Data Scheduling 9

2.1.1 Host Identity Protocol (HIP) 11

2.1.2 Multipath Packet Reordering Problem 13

2.1.3 TCP-friendliness and Fairness of the Multipath Congestion Control 14

2.1.4 Game-Theoretic Approach in Multipath Network Sharing 16

2.2 Real-time Multimedia Data Transmission 19

2.2.1 Internet Video Multicasting and Broacasting 19

2.2.2 Adaptive Hybrid Error Correction (AHEC) 20

2.2.3 Redundancy Information 21

2.2.4 Multicast Data Dissemination 23

2.2.5 Scalability of Multicast Trees 24

2.2.6 Topological Characteristics of Network Nodes 25

2.2.7 Optimal Positioning of the Special Network Nodes . 26

3 Summary of the results **27**

3.1 Multipath HIP 27

3.1.1 Multipath Data Scheduler 28

3.1.2 mHIP Linux Implementation 29

3.1.3 Some Experimental Results 31

3.2 Improving TCP-friendliness and Fairness for mHIP 35

3.2.1 Two-level Congestion Control for mHIP 35

3.2.2 Balancing between Aggressiveness and Responsiveness 36
3.2.3 Experimental Validation 37
3.3 Game-theoretic Approach in Multipath Network Sharing . . 39
3.3.1 Multiuser Multipath Network Routing Game 40
3.3.2 Routing Game with Traffic Delay Function $1 - e^{-\alpha_e \delta_e(x)}$ 41
3.3.3 Experimental Example 43
3.4 Improving Scalability of Real-time Data Transmission . . . 45
3.4.1 Control Nodes and Regions 45
3.4.2 Control Nodes Assignment Algorithm 46
3.4.3 Redundancy Optimization 47
3.4.4 Empirical Evaluation 51
3.5 Mediating Multimedia Traffic With Strict Delivery Constraints 53
3.5.1 Problem Statement 53
3.5.2 Node Characteristics 54
3.5.3 Node Location Assignment 56
3.5.4 Example Application Scenarios 57

4 Conclusions and Future Work 59

References 61

Publications 75

List of Publications

1. A. Gurtov, T. Polishchuk, *Secure Multipath Transport for Legacy Internet Applications.* In Proceedings of the Sixth International Conference on Broadband Communications, Networks, and Systems (BROADNETS 2009), September.

2. T. Polishchuk, A. Gurtov, *Improving TCP-friendliness and Fairness for mHIP.* In Infocommunications journal, 2011, Volume III/I, pages 26-34.

3. J. Chuyko, T. Polishchuk, V. Mazalov, A. Gurtov, *Wardrop Equilibria and Price of Anarchy in Multipath Routing Games with Elastic Traffic.* In International Journal of Mathematics, Game Theory and Algebra, 2012, Volume 20, Issue 4.

4. T. Polishchuk, M. Karl, T. Herfet, A. Gurtov, *Scalable Architecture for Multimedia Multicast Internet Applications.* In Proceedings of the 13th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2012), June.

5. M. Karl, T. Polishchuk, T. Herfet, A. Gurtov, *Mediating Multimedia Traffic With Strict Delivery Constraints.* In Proceedings of the IEEE International Symposium on Multimedia (ISM 2012), December.

Chapter 1

Introduction

The quickly growing community of Internet users is requesting multiple services. The QoS requirements may include delay, bandwidth, cost, packet loss, etc., and the whole system including content providers, network operators, service providers, device manufacturers and technology providers need to ensure that these demands can be met. From the performance point of view, there is a tight interplay between the application and the network design. The network must have adequate performance to support the target application.

In this thesis we consider how to improve the quality of users' experience concentrating on two popular and resource-consuming types of applications: bulk data transfer and real-time video streaming. We share our view on the techniques which enable feasibility and deployability of the network functionality leading to unquestionable performance improvement for the corresponding applications.

The Internet accommodates many types of traffic. Following [153] we differentiate between the real-time and elastic traffic. Elastic traffic refers to that of applications where the transmitted information is not time-sensitive, but requires eventual correct delivery. Examples of applications that generate elastic traffic are email, web-browsing, file transfers (FTP), Telnet, and any application that works without timely delivery. The Internet copes with elastic traffic very well. Protocols like TCP control the transmission rate of elastic traffic and allow for reliable transmission.

Real-time traffic refers to that of applications where the transmitted information is only useful if it is received within a small delay. Examples of applications that generate real-time traffic are voice over IP (VoIP), IPTV, video conferences, online gaming and generally any application that requires small end-to-end delay. The telecommunications market is expanding rapidly. It is foreseen that next-generation systems will have to support applications with increased complexity and tighter performance requirements. But the current Internet does not cater well to the time constraints imposed by such real-time applications. There is significant interest in developing the Internet that can accommodate real-time traffic.

Different applications react differently to the starvation of resources. A real-time multimedia stream may be completely unable to decode meaningful content, whereas a file download can be delayed slightly. Hence, choosing the optimization strategies in order to improve the service quality, we take in consideration the underlying application.

This work studies both types of traffic. Even though we aim to improve application data throughput in both cases, the methods we propose to achieve better performance are different. Our multipath solution improves data throughput on the wedge layer below transport, which means that any kind of traffic should be able to benefit from the ability of the protocol to use the underlying paths diversity. But most of the experiments in the first part of the thesis are conducted with the use of TCP data, that relates to the elastic traffic type. In the later part of the thesis the focus is shifted to the real-time latency-sensitive traffic. We suggest an architecture, which should meet the strict loss rate and delay requirements of the multimedia multicast applications. In this case the throughput improvement is achieved as a result of redundancy optimization.

Another important aspect is heterogeneous nature of the current and future Internet. A growing deployment of different technologies: WLAN, HSDPA, Bluetooth, WiMax, 4G, etc. leads to highly heterogeneous networks. The high variation of link characteristics makes it more and more challenging to provide stable end-to-end data transmission. In order to enable multipath data transmission we utilize a careful estimation of the end-to-end path capacities to be able to effectively balance the load between multiple paths and provide timely data delivery, and in the design of mHIP we include path probing to supply the sender with the fresh path characteristics. For more effective multicast multimedia data dissemination we propose to meet this challenge by protecting the data using adaptive hybrid error correction (AHEC), and serving the regions with similar link characteristics separately in order to optimize error correction techniques used on each region. Our new scalable multicast overlay architecture noticeably reduces the network load in real-time multicast scenarios with heterogeneous network structures.

Clearly both multipath data transmission and multicast content dissemination are the future Internet trends. In this thesis we discuss problems related to the deployment of the described ideas.

1.1 Research Problems and Scope

"Quality improvement is a never-ending journey. " (Tom Peters)

Development and evolution of Internet applications calls for new efficient methods to support advanced network capabilities such as network resource management, data provisioning and protection. In this thesis we propose to improve the quality of users' experience by dealing with the applications and utilizing both elastic and real-time types of network traffic. We design architectures and propose techniques to enable feasibility and deployability of the network functionality leading to the performance improvement for the corresponding applications. The manuscript consists of five publications, which cover multiple problems related to the deployment of these methods.

In order to boost progressive file download we suggest to use multiple available paths between the server and the client simultaneously inside one end-to-end connection. In Publication I we design an extension for Host Identity Protocol (HIP), and propose a multipath datscheduling solution on a wedge layer between IP and tra-ansport. Optimization addresses the way in which the data packets are distributed between the paths depending on their characteristics in order to achieve the best possible data throughput.

In Publication II we propose a TCP-friendly congestion control scheme for the mHIP secure multipath scheduling solution. We enable two-level control over aggressiveness of the multipath flows to prevent stealing bandwidth from the traditional transport connections in the shared bottleneck. We demonstrate how to achieve a desired level of friendliness at the expense of inessential performance degradation.

In Publication III we apply game-theoretic analytical modelling to solve the fair resource allocation problem. We investigate how the multihomed multipath-enabled hosts coexist in the same network and attempt to answer the following questions: Do the multiple HIP users with selfish objectives each exploiting similar scheduling techniques share multipath network fairly? Is such a network sharing optimal or does it need to be improved by applying some global congestion controllers on a higher level?

Targeting real-time multimedia applications with the specific constraints unveils the condition of taking care of timely delivery and meeting the residual loss requirement. Elastic traffic can tolerate fluctuations during transmission up to a certain degree, while real-time flow transmission is characterized by the upper error-rate and delay limits. Error-prone network segments can be effectively protected by the new error-correction schemes, e.g. hybrid error correction (HEC) frameworks.

In Publication IV we design a scalable multicast architecture for potentially large overlay networks. Our techniques address suboptimality of the adaptive hybrid error correction (AHEC) scheme in the multicast scenarios. A hierarchical multi-stage multicast tree topology is constructed in order to improve performance of AHEC and guarantee QoS for the multicast clients. The architecture has to adhere to the two strict constraints defined by the application: maximum allowed retransmission delay and target error rate at the receivers. Using analysis and simulations we prove that our multi-stage multicast architecture significantly reduces the amount of redundancy information introduced into the network and brings it closer to the Shannon bound.

Furthermore, the actual network topologies contain multiple different types of physical transmission channels like Ethernet, WLAN or 4G links. The error-correction mechanism must be adjusted individually to the different parts of the network in order to reduce the amount of redundantly sent data and provide a satisfying experience at the receiver at the same time. In Publication V we propose to reduce network load by tailoring error-correction schemes to both their application scope and underlying network topology. We introduce the idea how to exploit parallel networks by including the supplementary data from them into the primary multicast network, if the appropriate content is available. It leads to a relief of traffic in parts of the network. We propose to implement these functions as operating modes of the multi-purpose nodes, which we call Mediators following their operating principle of mediating traffic between multiple network segments. Thereby, Mediator nodes are introduced into the network where it is appropriate, and divide each end-to-end path between the source and receivers into several segments. This way non-error-prone links are released from carrying redundant data required by error-prone links as it happens in traditional end-to-end environments.

1.2 Research History

I started the work on enabling multipath functionality for the Host Identity Protocol as a part of the 3-year FISHOK project in 2008. The topic was introduced by my instructor, the head of HIIT Networking group Andrei Gurtov. Multipath routing is an active area of research. Despite the fact that several techniques of utilizing path diversity on the different layers of the TCP/IP stack have been proposed, multipath routing was not yet deployed in practice. We presented the initial design of an online multipath scheduling algorithm for HIP at the BROADNETS 2009 conference.

In 2009 I visited Trilogy summer school, where I presented a mHIP poster and discussed the proposal with several researchers working on the creation of different multipath-enabled protocols. This gave me some new research ideas and the motivation for the development of the TCP-friendly congestion control for mHIP, in order to prove its ability to compete with the other multipath proposals. The results were presented at the ACCESSNETS'10 and later extended for a journal (Publication II).

At the end of 2009 I met Julia Chuiko from the Karelian Research Center of Russian Academy of Sciences, who was visiting HIIT Networking group. Her presentation on the Wardrop equilibria and price of stability for bottleneck games with splittable traffic and the follow-up discussions gave a birth to the idea of applying the game-theoretical methods to the design of the fair congestion control for mHIP. Working together on the problem we were able to create an appropriate model for splitting TCP traffic to multiple paths inside HIP, and answer several interesting research questions related to the multipath fair resource allocation. The results of this work were first presented at the GTM'10 as an extended abstract and later published as a journal article (Publication III).

In the beginning of 2011 I spent two months in Saarbrucken University visiting the Telecommunications Group headed by Thorsten Herfet. There I became interested in the topic of multicast and broadcasting of multimedia data and optimization of redundancy in the networks protected by the Hybrid Error Correction. Working together with a PhD researcher, Michael Karl, I proposed a scalable multicast architecture which allows to increase throughput of the real-time multimedia data dissemination. Later Michael paid a couple of short visits to HIIT, and we presented the work in Publication IV. Additionally Michael proposed to use Mediators to futher improve scalability and performance of the multicast and broadcast networks. The results of this joint work were finalized in Publication V.

The completion of this thesis would not be possible without support of the Future Internet Graduate School during the last year of my studies.

1.3 Contributions

I actively participated in writing of all the papers presented in this thesis. My main contributions per publication are highlighted below.

Publication I: The idea of the design and general motivation of the paper were initiated by my co-author and instructor Andrei. I studied the related literature, elaborated his design ideas and proposed the online algorithm for multipath data scheduling. Using analysis and simulation I proved the efficiency of the proposed scheme, which I estimate to about 70% of the total work.

Publication II: I conducted most of the work for this publication, including design, implementation, experimentation and writing (about 90%), while the motivation and technical advice were provided by my co-author.

Publication III: The theoretical modelling and analysis originated from my co-authors. I wrote the introduction and related work for the paper, as well as conducted and desribed the experiments, which I estimate to about 40% of the total work.

Publication IV: I presented the main idea of this paper to my co-authors and did most of the work (about 80%), including implementation, experimentation and writing. My co-authors kindly provided the technical support and advise, as well as helped to evaluate the novelty and correctness of the proposed solution.

Publication V: In this paper we extend the idea presented in Publication IV. The design and experimental part of this publication belong to the first author. I contributed to the related work, problem statement, location assignment algorithms and example application scenarios sections (about 30% of the work).

1.4 Structure of the Thesis

Chapter 2 presents the background and related work. It also contains a detailed description of the main concepts, formulas and definitions used throughout the thesis. Chapter 3 discusses the main findings of the publications at a high level, as well as presents some practical examples, while much of the experimental results and technical details are not repeated. Finally, the work is concluded in Chapter 4.

Chapter 2

Literature Review and Background

In this chapter we outline the related work and review the preliminaries needed for better understanding of the results presented later in this thesis.

2.1 Multipath Data Scheduling

Modern mobile devices, equipped with several network interfaces, as well as multihomed residential Internet hosts are capable of maintaining multiple simultaneous attachments to the network.

In recent years, there have been a number of efforts within the networking community to enable data transmission over multiple paths on different layers of the TCP/IP stack. In some schemes, such as SCTP [67], MPTCP [13], pTCP [63], mTCP [162], R-MTP [100], MPRTP[138] multiaddressing support is provided on transport or network [117], [23], [77], [35], [74] layers. Multiple interfaces of the same technology can also be striped at the link layer. Several solutions propose bandwidth aggregation on the application layer [46], [128]. An alternative approach, implemented in HIP [105], LIN6 [66], MAST [28], MIP6 [80] is to conduct multiaddressing support in a functional layer between IP and transport.

The main idea of bandwidth aggregation on the link layer is to stripe data across a bundle of physical channels, as it was done in [4], [139]. A method for channel aggregation in cellular networks is described in [24]. Another interesting approach [133] targets WLAN users who should be able to split their traffic among several available access points. However, the link layer has no notion of IP addresses and striping solutions on the link layer are only applicable to the links with equal technology.

Placing multipath functionality on the lower layers enables efficient utilization of a particular link type and presents more generic solution for all the upper-layer protocols and applications. On the other hand, solutions on upper layers are better tuned for the needs of a specific application and could be implemented more easily.

The advantages of network layer approaches are that they are relatively easy to deploy, totally transparent to applications and involve only minimal changes in the infrastructure in contrary to the transport-layer solutions; but are not able to cope with packet reordering problem and do not generally support proper per-flow congestion control, which is needed to provide the required level of TCP-friendliness to the external connections.

The transport layer can naturally obtain information on the quality of different paths. For example, SCTP [142] can perform measurements across several paths simultaneously, and then map flows to one or another path. TCP-MH [85] can detect when the current path has stopped working well, for instance, if the frequency of repetition becomes too high, and decide to try another path.

SCTP protocol [142] supports a notion of multiple paths for fault-tolerance. Concurrent multipath transport (CMT) extension for SCTP [67] enables hosts to use multiple independent paths simultaneously. Although implemented in several operating systems, SCTP is not widely used mainly because application developers need to change their applications to use SCTP.

TCP [141] was designed for machines that utilize a single path for communication, which eventually lead to designing its multipath variant MPTCP [38]. In TCP, connection-specific functions, such as flow control and connection establishment are tightly coupled with path-specific functions such as Maximum Transmission Unit (MTU) discovery, congestion avoidance and retransmissions. Implementing multipath transfer in TCP requires significant re-factoring of the code, separating connection and path-specific components so that functions such as congestion control could be implemented per each path.

Wedge-layer approaches have an advantage of being able to maintain multiaddressing information across transport associations. For instance, the HIP multihoming feature [114] provides multiaddressing support for HIP-enabled hosts and explores path diversity. Transport activity between two endpoints may well be able to use multiaddressing immediately and with no further administrative overhead. Moreover, wedge-based locator exchange protocols can be incorporated without necessitating modification to any host's IP or transport modules.

A number of applications or transport connections can be allocated independently to different paths [123]. As an example, popular web browsers open several parallel TCP connections to download a page. Such an approach avoids complications resulting from spreading packets from a single transport connection over multiple paths. However, it has an obvious drawback – if there are fewer active bulk transport connections than links, it is not possible to utilize all available paths. Simultaneous Multiaccess (SIMA) [124] implements such an approach using flow-binding extensions for HIP.

Several proposals in the related work assume the presence of a proxy in the network that can serve as a termination point of multipath TCP extensions [18]. Such an approach works only for plain-text TCP communication and fails in the presence of IPsec [11] encryption or authentication mechanisms. When TCP packets are protected with IPsec, the proxy is unable to observe or modify the packets. Therefore, if HIP is used end-to-end, proxy-based solutions are not applicable.

2.1.1 Host Identity Protocol (HIP)

The Host Identity Protocol (HIP) [50], [51], [58], [105], [106], [113] was proposed to overcome the problem of using IP addresses simultaneously for host identification and routing. The idea behind HIP is based on decoupling the network layer from the higher layers in the protocol stack architecture (see Figure 2.1). HIP defines a new global name space, the Host Identity name space, thereby splitting the double meaning of IP addresses. When HIP is used, upper layers do not rely on IP addresses as host names any more. Instead, Host Identities (HI) are used in the transport protocol headers for establishing connections. IP addresses at the same time act purely as locators for routing packets towards the destination. For compatibility with IPv6 legacy applications, Host Identity is represented by a 128-bit long hash, the Host Identity Tag (HIT) [59].

HIP offers several benefits including end-to-end security, resistance to CPU and memory exhausting denial-of-service (DoS) attacks, NAT traversal [86], [143], rendezvous services [90], mobility and multihoming support [53] and other services [91].

To start communicating through HIP, two hosts must establish a HIP association. This process is known as the HIP Base Exchange (BEX) and it consists of four messages transferred between the initiator and the responder. After BEX is successfully completed, both hosts are confident that private keys corresponding to Host Identifiers (public keys) are indeed possessed by their peers. Another purpose of the HIP base exchange is to create a pair of IPsec Encapsulated Security Payload (ESP) [71] Security

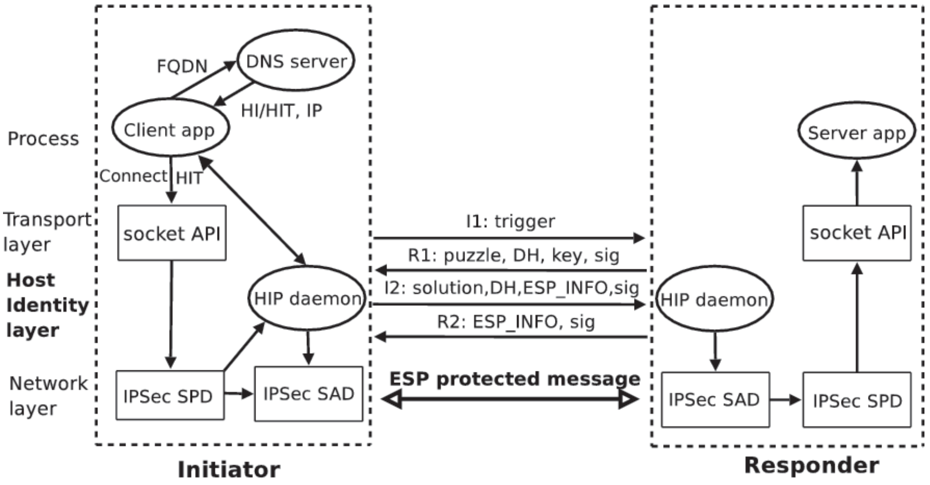

Figure 2.1: HIP architecture.

Associations (SAs), one for each direction. All subsequent traffic between communicating parts is protected by IPsec. A new IPsec ESP mode, Bound End-to-end Tunnel (BEET) [116] is used in HIP. The main advantage of the BEET mode is low overhead in contrast to the regular tunnel mode.

Figure 2.1 illustrates the overall HIP architecture including the BEX [57]. The initiator may retrieve the HI/HIT of the responder from a DNS directory [115] by sending a FQDN in a DNS query. Instead of resolving the FQDN to an IP address, the DNS server replies with a HI. The transport layer creates a packet with the HI as the destination identifier. During the next step HI is mapped to an IP address by the HIP daemon on the Host Identity layer. The packet is processed in the network layer and routed to the responder. As a result, the conventional 5-tuple socket becomes {protocol, source HI, source port, destination HI, destination port}.

Since neither transport layer connections nor security associations (SAs) created after the HIP base exchange are bound to IP addresses, a mobile client can change its IP address (i.e., upon moving, due to a DHCP lease or IPv6 router advertisement) and continue transmitting ESP-protected packets to its peer. HIP supports such mobility events by implementing an end-to-end three-way signaling mechanism [114] between communicating nodes. HIP multihoming uses the same mechanisms as mobility for updating the peer with the current set of IP addresses of the host. It provides multiaddressing support in a functional layer between IP and transport.

Multihoming and advanced security features make the Host Identity Protocol a good candidate to provide multipath data delivery.

2.1.2 Multipath Packet Reordering Problem

When data packets are sent over several paths inside one connection they can experience different propagation delays and arrive out of order. A significant amount of related work is devoted to measurement of packet reordering and analysis of its influence on the performance of protocols on different layers of the TCP/IP stack and corresponding applications [36], [47], [78], [99], [125], [126].

TCP implicitly assumes corruption-free links along the entire path, therefore packet loss or corruption is often mistaken for an indication of path congestion. As a consequence, TCP may not be able to fully utilize its fair share of the available path capacity. But not only TCP, which takes reordering as a sign of congestion, results in degraded performance. Even some UDP-based applications such as VoIP [92] are sensitive to packet reordering.

Multipath traffic splitting algorithms could minimize reordering of the packets arriving from parallel paths. In datagram networks, a few proposals for traffic splitting forward packets onto multiple paths using a form of weighted round-robin or deficit round-robin [136], [160] scheduling. These schemes cause significant packet reordering and thus are not used in practice. Alternative schemes avoid packet reordering by consistently mapping packets from the same flow to the same path. Commercial routers implement the Equal-Cost Multipath (ECMP) [62] feature of routing protocols such as OSPF and IS-IS. Hash-based versions of ECMP divide the hash space into equal-size partitions corresponding to the outbound paths, hash packets based on their endpoint information, and forward them on the path whose boundaries envelop the packet's hash value [21], [112]. Though these schemes provide a good performance when operating with static load balancing, they are unsuitable for the emergent dynamic load balancing protocols [73], [154] where they may overshoot the desired load by as much as 60% [74]. A few papers analyze the performance of various splitting schemes. Cao et al. [21] evaluated the performance of a number of hashing functions used in traffic splitting. Rost and Balakrishnan [131] evaluated different traffic splitting policies, including rate-adaptive splitting methods.

But despite all the attempts to prevent packet reordering in the scenarios when multiple paths with variable delays on the links are used simultaneously, eventual out-of-order delivery is almost inevitable [48]. The TCP receiver sends duplicate acknowledgements (*dupacks*) to the sender, which will falsely indicate packet loss. It can lead to unnecessary retransmissions and a substantial reduction of the congestion window thereby reducing total throughput.

The authors of [94] surveyed and analyzed relevant techniques on coping with multipath TCP packet reordering. Several improvements for TCP do exist that make the reordering tolerable including the Eifel algorithm [49], [98], TCP-NCR[14] and DSACK [163]. The Eifel response algorithm restores the congestion state to the state before entering the (unnecessary) loss recovery.

Several proposals (e.g. [15], [163]) suggested to increase the *dupthresh* value defining the number of *dupacks* which serve as an indication of congestion, as a cure for the mild packet reordering. Compared with the default *dupthresh* of three, the proposed techniques improves connection throughput by reducing the number of unnecessary retransmissions. But one should adjust the *dupthresh* value carefully since making it too large slows down the reaction of the system to the actual losses and can significantly degrade the overall performance in the networks with high loss rates.

Non-Congestion Robustness for TCP (TCP-NCR) helps to better disambiguate segment loss from reordering. Since three duplicate ACKs may not be sufficient to distinguish loss from reordering, TCP-NCR uses a dynamic DupACK threshold to a value that approximates a congestion window of data having left the network (which corresponds to one round-trip time). A relevant study of the impact of packet reordering on modern TCP variants, including TCP-NCR, is presented by Feng [36]. They conclude that existing reordering-tolerant algorithms can significantly improve the performance of TCP.

Using the methods described in [14], [15], [98] and [163], we suggest the improvement for multipath HIP, which reduces the level of reordering on the receiver and significantly improves the TCP-friendliness of our scheme.

2.1.3 TCP-friendliness and Fairness of the Multipath Congestion Control

Multipath-enabled protocols are usually designed to be able to shift their traffic from congested paths to uncongested regions in order to balance the load and better utilise the available Internet capacity. Multipath congestion control should be designed to balance the load in order to avoid congestion hotspots. Additionally it should take care of the fair resource allocation.

TCP traffic comprises a major share of the total Internet traffic. Proper per-flow congestion control is required to limit aggressiveness of the proposed multipath solutions [140].

TCP-friendliness [152] has emerged as a measure of correctness in Internet congestion control. The notion of TCP-friendliness was introduced to restrict non-TCP flows from exceeding the bandwidth of a conforming

TCP running under comparable conditions. Protocols commonly meet this requirement by using some form of AIMD (Additive Increase Multiplicative Decrease) [25] congestion window management, or by computing a transmission rate based on equations derived from an AIMD model [8], [9], [12]. In [43] the author studied the impact of feedback modeling on the Internet congestion control, and the problem of convergence of binary adjustment algorithms to fairness. It was proved in [44] that the AIMD model offers the best trade-off between smoothness and responsiveness and offers the fastest fairing, which provided a theoretical justification for using AIMD in TCP congestion avoidance [68]. Although TCP is not the only possible congestion control mechanism in the Internet [52], in this thesis we limit ourselves to TCP as the most commonly used protocol at the moment.

Definitions

TCP-friendliness is a generic term describing a scheme that aims to use no more bandwidth than TCP uses. In this thesis we study mHIP congestion control in view of the criteria proposed in [152]:

A *TCP-compatible* flow, in the steady state, should use no more bandwidth than a TCP flow under comparable conditions, such as packet-loss rate and round-trip time (RTT). However, a TCP-compatible congestion control scheme is not preferred if it always offers far lower throughput than a TCP flow.

A *TCP-equivalent* scheme merely ensures the same throughput as TCP when they experience identical network conditions. Although a TCP-equivalent scheme consumes TCP-equivalent bandwidth when working by itself, it may not coexist well with TCP in the Internet.

TCP-equal share is a more realistic but more challenging criterion than TCP-equivalence and states that a flow should have the same throughput as TCP if competing with TCP for the same bottleneck. A TCP-equivalent flow may not be TCP-equal share, but the opposite is always true.

To be able to meet all three criteria a TCP-friendly scheme should use the same bandwidth as TCP in a steady-state region, while being aggressive enough to capture the available bandwidth and being responsive enough to protect itself from congestion, as the packet-loss condition changes in the paths in the transient state. *Aggressiveness* of a scheme describes how the scheme increases the throughput of a flow before encountering the next packet loss, while *responsiveness* describes how the scheme decreases the throughput of a flow when the packet-loss condition becomes severe.

To quantify friendliness of a protocol respect to the standard TCP, we introduce the factor of friendliness metric:

$$FF(flow) = \frac{T(flow)}{T(TCP)}$$

Here $T(\cdot)$ denotes the average flow throughput in Mbps. $FF = 1$ indicates the solution satisfies the strongest TCP-equal share criterion, while a solution resulting in $FF > 1$ is more aggressive than a typical TCP and the one with $FF < 1$ may not be TCP-compatible.

According to the resource pooling principle [156] when several subflows of one connection share a bottleneck, their resource consumption adds up. Multipath connections with a large number of TCP-friendly subflows can compete unfairly against a smaller number of regular TCP connections. Each subflow is as aggressive as a single TCP, and a bundle of n TCP-friendly subflows will hence use an approximately n times greater share of the bottleneck resource than they should. *TCP-fair* multipath connection should displace no more TCP traffic than a traditional TCP stream would displace. A number of methods [54], [55], [56], [65], [79], [157] were proposed to study and solve the TCP-fairness problem for the multipath-enabled protocols.

The congestion control solution for mHIP, which we present further in this work, is also designed to meet the TCP-friendliness and TCP-fairness criteria.

2.1.4 Game-Theoretic Approach in Multipath Network Sharing

Game-theoretic frameworks are powerful in describing and analyzing competitive decision problems. Game theory has been used to study various communication and networking problems including routing, service provisioning, flow-rate controlling by formulating them as either cooperative or non-cooperative games. The authors of [10] summarized different modelling and solution concepts of networking games, as well as a number of different applications in telecommunication technology.

Networking games have been studied in the context of road traffic since 1950, when Wardrop proposed his definition of a stable traffic flow on a transportation network [155]. Both *Wardrop* and *Nash* [111] *equilibria* are traditionally used to give an idea of fair resource sharing between players [26], [39], [104]. However, they do not optimize social costs of the system. In 1999 the concept of the *price of anarchy* was proposed by Koutsoupias and Papadimitriou to solve this problem. In [89] network routing was modelled as a non-cooperative game and the worst-case ratio of the social welfare, achieved by a Nash equilibrium and by a socially optimal set of strategies. This concept has recently received considerable attention and is widely used to quantify the degradation in network performance due to unregulated traffic [103], [132].

In the conventional TCP/IP networking [141] multiple users share communication links and buffering capabilities of the network routers. When users do not cooperate and do not respect the protocol rules, it is possible that unfair or unstable behaviours emerge in the system. This problem of the TCP protocol has already been addressed in the networking literature using a game-theoretic perspective. For example, Nagle [110] and Garg et al. [40] proposed solutions based on creating incentive structures in the systems that discourage evil behaviour and show the potential applications of Game Theory within the problem of congestion control and routing in packet networks.

An excellent analysis of TCP behaviour in the context of Game Theory has been proposed by Akella et al. [7]. In this work, a combination of analyses and simulations is carried out in an attempt to characterize the performance of TCP in the presence of selfish users. Our results for multipath networks presented in this work agree with the main conclusions for the traditional unipath networks from [7]: when the users use TCP New Reno loss-recovery [37] in combination with drop-tail queue management the equilibrium strategies of the users are quite efficient for fair resource allocation.

Later P. Key et al. constructed a series of game-theoretical models to analyse the performance and benefits of implementing multipath routing and proposed several congestion control mechanisms aiming to optimize fairness in the multipath-enabled networks [17], [82], [83], [102].

Nash and Wardrop Equilibria

Let $x = (x_1, \dots, x_n)$ be the user's strategy profile. For the original profile x the new profile $(x_{-i}, x'_i) = (x_1, \dots, x_{i-1}, x'_i, x_{i+1}, \dots, x_n)$, where the user i changes his strategy from x_i to x'_i and all other users keep their strategies the same as in x.

Function $PC_i(x)$ defines the individual costs of i-th user. Each user i tries to minimize his individual costs $f_{ie}(x)$: $PC_i(x) = \max\limits_{e:x_{ie}>0} f_{ie}(x)$.

Definitions

A strategy profile x is a *Nash equilibrium* iff for each user i for any profile $x' = (x_{-i}, x'_i)$ holds $PC_i(x) \leq PC_i(x')$.

A strategy profile x is *a Wardrop equilibrium* iff for each i:
if $x_{ie} > 0$ then $f_{ie}(x) = \min_{l} f_{il}(x) = \lambda_i$ and if $x_{ie} = 0$ then $f_{ie}(x) \geq \lambda_i$.

Nash and Wardrop equilibria definitions are not always equivalent. It depends on the type of traffic delay functions defined in the model.

Properties

In Publication III we utilize the following well known properties of the Nash and Wardrop equilibria:

Property 1. *If the strategy profile x is a Wardrop equilibrium then x is a Nash equilibrium.*

Property 2. *If all delay functions $f_e(x)$ in the model are strictly increasing by all x_{ie} then in this model any Nash equilibrium is a Wardrop equilibrium.*

Property 2 means that it is always possible to redistribute some small user's traffic amount from any of routes to the less loaded routes in order to decrease traffic delay on this route for this user.

The Price of Anarchy

A strategy profile x is a *social optimum* if it provides a minimum of social costs by all the profiles. The social costs function is not convex, and its local minimum can differ from the the global optimum. But we can try to obtain some stationary points and check their optimality.

Price of Anarchy is a ratio of equilibrium social costs in the worst-case equilibrium and optimal social costs

$$PoA(\Gamma) = \max_{x \text{ is an equilibrium}} \frac{SC(x)}{SC_{opt}}.$$

Here the social optimum SC_{opt} is a solution of a minimization problem

$$SC(x) \to \min_{x \text{ is a strategy profile}}$$

The value is the same for any Wardrop equilibrium providing the Price of Anarchy cannot be infinite.

2.2 Real-time Multimedia Data Transmission

Efficient and reliable transport of multimedia content is a critical issue of today's IP network design. The amount of this traffic type highly increased in the last years and it continues to grow [27]. But not only typical multimedia traffic originated by web platforms like Youtube is dominating the IP-based world. Today Internet video occupies 50 percent of consumer Internet traffic, and will reach 62 percent by the end of 2015 [27].

Furthermore, it is expected that popular content is streamed not just to a single user, but to multiple users attempting to access the same content at the same time in the form of multicast or broadcast - point to multipoint (p-t-m) services, which target simultaneous distribution of multimedia content to many interested users.

Today's approaches to cope with this development are manifold.

2.2.1 Internet Video Multicasting and Broacasting

Peer-to-peer(P2P) [19], [135] networking turns out to be a good distribution approach for content valuable for a wide audience. Due to the agnostic overlay construction process, triggered by the users themselves, the ISPs lose control of the network transmissions and suffer from loss of revenue [6]. Other shortcomings are the highly heterogeneous consumer access bandwidth and high churn rates. The recently proposed Proactive Network Provider Participation for P2P (*P4P networks*) [159] try to overcome some of these drawbacks by introducing interaction between P2P networks and the network topologies but they also suffer from the aforementioned basic P2P challenges [109]. Another suitable approach to handle the increasing amount of data volumes are *Content Delivery Networks (CDNs)* [20]. Hereby, a cluster of surrogate servers, which are distributed across the network, is used to store copies of the original content in order to increase content delivery quality, speed and reliability [120]. CDN challenges are synchronization and updating the cached content within the delivery network. An approach that basically tries to combine broadband IP communication and broadcasting is Dynamic Broadcast [127]. Thereby, broadcasters can offer additional services over broadband connections to satisfy the consumer's needs. As a consequence, it will be possible to shift more content to the broadband which helps to save costs, especially if the audience is fairly small. This approach is currently influencing the Hybrid Broadcast Broadband TV standardization process [2]. Furthermore, in [108] the authors focus on broadcasting augmentation data for GPS, especially on the distribution of such data via IP datacasting.

2.2.2 Adaptive Hybrid Error Correction (AHEC)

A quickly growing interest in the real-time streaming and interactive applications leads to a higher stress on the networks. New transmission approaches have to be investigated to prevent the *Future Media Internet* from being impaired by non-essential traffic. Owing to error-prone network segments the use of new error-correction schemes are also required, e.g. *hybrid error correction* frameworks [148].

As it is well known, there are two basic categories of Erasure Error Recovery (EER) techniques: Automatic Repeat reQuest (ARQ) and Forward Error Correction (FEC) erasure coding [64]. The integrated FEC / ARQ schemes are referred to as *Hybrid Error Correction (HEC)* schemes. Several studies indicate that HEC schemes are much more efficient for recovering missing packets for multicast services than the schemes with either FEC or ARQ alone [33]. Using ARQ and Packet Repetition (PR) techniques, the authors of [151] developed a HEC-PR scheme for satisfying a certain packet loss rate (PLR) requirement under strict delay constraints and optimized its performance.

Later they discovered that better performance can be achieved by combining the HEC-PR with a traditional Type I HARQ scheme [149]. However, there was still a critical question to answer: Which scheme is optimal in a real-time media-based multicast scenario with guaranteeing a certain PLR requirement under strict delay constraints? To address the question the authors developed a general architecture of the Erasure Error Recovery (EER) combing all of the HEC schemes mentioned above into the Adaptive Hybrid Error Correction (AHEC) scheme [148]. Through optimizing the general architecture under strict delay constraints, the total needed Redundancy Information (RI) is minimized by choosing the best scheme automatically among the entire schemes included in the architecture.

The AHEC scheme was developed with the assumption of the Gilbert-Elliott (GE) erasure channel, basing on the studies [34], [42] showing that the simplified GE model is a very good approximation for the packet loss model in a wireless channel [84], [147]. AHEC operates according to the Predictable Reliability under Predictable Delay (PRPD) paradigm: based on a statistical channel model it adds the optimal amount of redundant information to meet the desired residual packet loss rate within the limited time constraint. The flexible combination of limited packet retransmissions issued by negative feedback and adaptive, packet-oriented FEC spans a large parameter space with few feasible configurations: the number of retransmission rounds and the FEC block length share the overall time budget.

Furthermore, Tan et al. optimized the performance of the AHEC scheme for DVB services over wireless home networks [151]. The authors analysed the needed redundancy information for the HEC-PR and HEC-RS cases of the general AHEC frameworks and showed how to minimize the RI value in multicasting scenarios with small group sizes (about 7 receivers). Later in [148] they noticed that in the general AHEC architecture, redundancy grows quickly with the increase of the group size if the size of the group is small (less than 20 receivers), but if the size of the group is large enough, the total needed redundancy will increase very slightly with the increase of the group size. They concluded that the AHEC scheme can be suitable for the multicast scenarios with large groups.

Previous work considered only limited multicast topologies with small groups of multicast receivers connected directly to the source. In the thesis we extend the multicast scenario to the more realistic Internet tree structure and study how to optimize AHEC for a wider range of multimedia multicast applications and bigger groups.

2.2.3 Redundancy Information

Further we review the definition and formulas for redundancy calculation, according to the framework provided in [151], which we use throughout our work described in Publications IV and V.

Definition

Redundancy information (RI) is the controlled redundancy added by the channel encoder and required to protect the receiver from any errors during data transmission.

In the general AHEC framework [150] the redundancy information consists of two parts. One part, denoted by r, is delivered with the main data block during the first transmission and it is produced by the FEC component of AHEC. The overall transmitted block length is calculated as $n = k + r$ for the data size of k and the redundancy amount of r. The second part of the total redundancy is the data, which restores lost packets after retransmissions. The number of retransmissions available is limited by the delay budget, which is left after the first transmission.

In this current work we focus on optimizing the second part of redundancy information produced by retransmissions, since optimization of FEC with hierarchical tree structures was already addressed in the related work [129], [134], [158].

Notation

RI_{HEC} - redundancy information introduced into the network by AHEC;
P_{target}, D_{target} - packet loss rate and delay requirements;
RTT_i, P_i - characteristics of the link;
RTT_{e2e}, P_{e2e} - characteristics of the end-to-end path between the source and receiver;
T_s - average interval between two continuous data packets;
N_T - the maximum possible number of transmissions for each data packet;
N_{rr} - the maximum possible number of retransmission rounds for each data packet;
$\tilde{N_{rr}} = \min(N_{rr}, N_T)$ - maximum allowable number of retransmissions (practical);
N^q - the number of transmissions of each missing data packet during q-th retransmission round;

Formulas for the RI_{HEC} calculation

The authors of [151] proposed the following formulas for calculation of the HEC redundancy information, which we use further in Publication IV and Section 3.4.3 of this thesis:

$$RTT_{e2e} = \sum_{i=1}^{N} RTT_i;\ P_{e2e} = 1 - \prod_{i=1}^{N}(1 - P_i)$$

$$N_T = \lceil \frac{\log(P_{target})}{\log(P_{e2e})} \rceil;\ N_{rr} = \lfloor \frac{D_{target} - \frac{RTT_{e2e}}{2} - (N_T - 1) * T_s}{RTT_{e2e} + T_s} \rfloor$$

$$RI_{HEC} = \sum_{q=1}^{\tilde{N_{rr}}} N^q P_{HEC}(q-1), \text{ where } P_{HEC}^{q-1} = P_{e2e}^{\sum\limits_{q=0}^{q-1} N^q}$$

The goal is to minimize the total needed redundancy information by optimizing the number of retransmission rounds needed to provide P_{target} without D_{target} violation:

$$RI_{HEC}^{opt} = \arg\min RI_{HEC}, \text{ s.t. } 1 \leq \tilde{N_{rr}} \leq N_{rr}$$

2.2.4 Multicast Data Dissemination

IP multicast [29],[30],[61] was first proposed more than two decades ago, as an efficient solution for dissemination of the same data from a single source to multiple receivers. It was deployed experimentally [119] but never adopted in any significant way by service providers. The failure of multicast [31] to achieve wide-spread can be explained by several technical and economic factors, including complexity of the multicast network management and uncertainty in how to appropriately charge for the service in the case when sources and receivers belong to different domains. As a result, for many years multicast was implemented only locally within the service providers' networks supporting IPTV [146] and conferencing applications, and also deployed in enterprise networks [76], where the aforementioned issues are mitigated.

Due to its unquestionable capability to significantly reduce network load, multicast remains the most studied research problem in computer networking [121]. According to [72] the main challenge in efficient information-centric network (ICN) design is how to build a multicast infrastructure that can scale to the general Internet and tolerate its failure modes while achieving both low latency and efficient use of resources. In topic-based ICNs, the number of topics is large while each topic may have only a few receivers [97]. IP multicast and application level multicast have scalability and efficiency limitations under such conditions. In IP multicast the amount of routing state is proportional to the number of multicast groups. To address this issue several multicast routing and forwarding proposals [70], [72], [130], [161], introduced the idea of using *Bloom filters* (*BF*) [16] in packet headers. This way the intermediate routers are purged from the burden of keeping states, leaving space for scalability improvement.

Internet Protocol Television (IPTV) has emerged as a new delivery method for TV [22]. In contrast with native broadcast in the traditional cable and satellite TV system, video streams in IPTV are encoded in IP packets and distributed using IP multicast or unicast. IPTV has an advantage over satellite, terrestrial or cable TV because it has an embedded return channel, which enables a service provider to add more interactivity. Another advantage is that the service provider can combine TV, phone and Internet access into one network, which would decrease the deployment cost significantly.

Wireless 802.11 networks, typically used as the last hop of an IPTV network, would add mobility, which is of great value for hot-spot deployment and is often required for in-home content distribution. However, wireless channels are typically unreliable, they suffer from interference and multipath fading, which cause packet bursts, losses and finally contribute to the degradation of the quality of users' experience. Multimedia Broadcast Multicast Service (MBMS), the point-to-multipoint feature, has been specified by the 3rd Generation Partnership Project (3GPP) in order to meet the increasing demands of multimedia download and streaming applications in mobile scenarios [5]. To overcome the low reliability of wireless networks, several proposals of additional error correction schemes have been considered [145]. As a suitable trade-off between easy dissemination and sufficient performance 3GPP introduced application layer FEC using Raptor Codes [41] as an additional means to provide reliability. H.264/AVC [60] is integrated into MBMS for encoding of broadcast applications.

An alternative design for IPTV multicast over wireless LAN using the Merged Hybrid Adaptive FEC and ARG (MHARQ) was proposed in [101]. MHARQ combines the advantages of receiver-driven staggered FEC and hybrid ARQ schemes to compensate the large dynamic range of WLAN channels and to achieve high reliability, scalability and wireless bandwidth efficiency for video multicast. The authors proposed a channel estimation algorithm for a receiver to dynamically determine the delayed FEC multicast groups to join and/or send ARQ NACK to request for retransmission.

2.2.5 Scalability of Multicast Trees

The idea of optimizing performance of an error correction scheme by using a hierarchical tree structure was first introduced by Radha and Wu in [129] and [158]. They developed a recursively optimal scheme for the placement of a given number of network-embedded FEC codecs within a randomly generated multicast tree with known link loss rates.

Shan et al. proposed in [134] an overlay multi-hop FEC (OM-FEC) scheme that provides FEC encoding and decoding capabilities at the intermediate nodes in the overlay path. Based on the network conditions, the end-to-end overlay path is dynamically partitioned into segments and appropriate FEC codes are applied over those segments.

Kopparty et al. [87] and Paul et al. [122] introduced the idea of optimizing the lengths of retransmission rounds in the design of transport multicast protocols (SplitTCP and RMTP), by allowing buffering at some intermediate nodes.

Scalability of the multicast dissemination trees controlled by the hybrid error correction has never been addressed before, to the best of our knowledge. Later in this work we present a *scalable multicast multistage architecture* aiming at optimizing performance of the AHEC scheme in order to serve predictable reliability for the needs of multimedia applications.

2.2.6 Topological Characteristics of Network Nodes

In Publication V we introduce multi-purpose relay nodes called Mediators into several positions within the tree networks typical for multicasting and broadcasting scenarios. We are utilizing the error-correction domain separation paradigm [75] in combination with selective insertion of the supplementary data from parallel networks, when the corresponding content is available.

The Mediator location assignment process is highly important since it determines where the actual error-correction takes place or where additional data is injected from the parallel networks. In the following the basic ideas of how to select suitable Mediator locations within the data dissemination network basing on the underlying network topology are presented.

Potential positions for the Mediator nodes could be identified using the topological characteristics of the network nodes, such as the (weighted) degree, centrality or betweenness of individual network nodes [118]:

Degree. The basic factor to determine how many adjacencies the node has is its degree. Obviously, the higher the degree the more connections to other node are available. This may lead to a definite occurrence in distribution tree structures. Thus, the degree of node n_i with adjacency matrix x_{ij} is defined as

$$degree(i) = \sum_{j}^{N} x_{ij}$$

where N represents the total number of nodes.

Closeness. Incorporating shortest-path measurements in the network leads to the closeness metric. Thereby, it is assumed that a larger path introduces more costs for interaction. The longer the paths to the other nodes, the smaller the metric value. The inverse of the sum of distances from one node n_i to all others is defined as the closeness centrality metric:

$$closeness(i) = \left[\sum_{j}^{N} d(i,j) \right]^{-1}$$

where $d(i,j)$ represents the distance between two nodes n_i and n_j in terms of hops, and N again reflects the total number of nodes.

Betweenness. Combining the number of shortest paths between any two nodes and the number of these shortest paths that pass a node n_i leads to the betweenness metric. Again, the more paths exploit this node, the higher the metric value. Let sp_{jk} represent the total number of shortest paths in terms of hops and $sp_{jk}(i)$ denote the number of paths passing node n_i, it holds for the betweenness metric:

$$betweenness(i) = \frac{sp_{jk}(i)}{sp_{jk}}$$

Obviously, this list is not exhaustive and can be extended by more characteristics from graph theory or traffic engineering.

2.2.7 Optimal Positioning of the Special Network Nodes

Further we review several assignment algorithms, which could be applied for optimization of the Mediator nodes positioning.

Li et al. [95] proposed an algorithm for finding the optimal placement of M web proxies among N potential sites under a given traffic pattern. The algorithm obtains the optimal solution for the tree topology using $O(N^3M^2)$ time. It works for the scenarios where the clients can request data only from the parent, but not sibling proxy. This model is applicable for the multicast multimedia data transmission within the tree dissemination networks as well.

For the general graph topologies a web server replicas placement model was proposed in Qiu et al. [127]. The authors formulated the problem as Minimum K-Median Problem, which is known to be NP-hard, and analyzed several approximation algorithms. They showed that a simple greedy approach, where the optimal locations for web replicas are chosen one by one according to the associated costs until M is reached, performs the best with the median performance within the factor 1.1-1.5 of optimal.

In Publication IV we proposed a control node assignment algorithm, which finds the optimal number of error-correcting relays and their recommended placements within the multicast dissemination tree topologies in $O(NlogN)$ time. The objective is to optimize redundancy information introduced into the network when HEC is applied. This algorithm is directly applicable for Mediator nodes location when they perform in the error-correcting relay mode, as described in Publication V.

Chapter 3

Summary of the results

In this chapter we summarise our results published in the journals and conference proceedings. More detailed analysis of the results is presented in the corresponding publications attached at the end of the thesis.

3.1 Multipath HIP

In Publication I we design an extension for Host Identity Protocol (mHIP), and propose a multipath data scheduling solution on a wedge-layer between IP and transport as shown in Fig. 3.1.

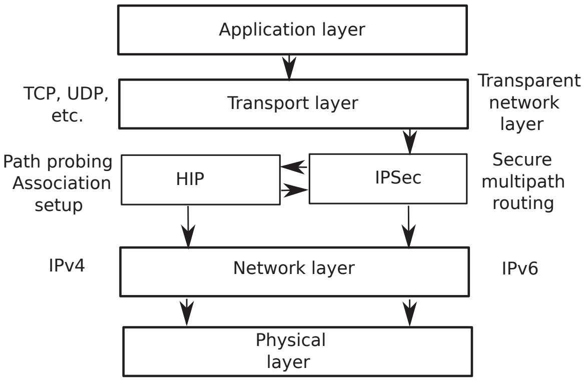

Figure 3.1: Position of mHIP in the TCP/IP stack.

Compared to the most relevant multipath transport layer proposals (e.g. [13], [67]) multipath HIP (mHIP) has multiple advantages. HIP multihoming feature automatically provides multiaddressing support for HIP-enabled hosts and explores path diversity. mHIP naturally solves the tasks which are challenging for any multipath design. These include providing end-to-end security for each individual multipath flow, facilitating the ability to traverse NATs and middleboxes, as well as mobility support, which are inherited from the standard HIP protocol implementation.

We take a different approach compared to the extending transport protocols. Since multipath functionality requires proper security mechanisms to avoid accepting packets from spoofed IP addresses, it is logical to implement it on the HIP [105] layer. HIP shields the presence of multiple paths from transport and application layers, presenting only the identity of the peer host, a Host Identity Tag (HIT). Therefore, all multipath functionality must be located below the HIP layer as the upper protocol layers only see a single path through HIT.

3.1.1 Multipath Data Scheduler

The scheduler located below the HIP protocol maintains an estimate of each path parameters, including the congestion window, retransmission timer, and MTU. It spreads packets from TCP connections over available paths proportionally to their capacity.

According to the resource pooling principle specified in [156] ideally the overall throughput is the sum of all link bandwidths. Since TCP or SCTP at the transport layer can not robustly differentiate between packet reordering and packet loss, the scheduler must minimize reordering at the receiver.

We formulate the multipath scheduling problem as an online optimization problem; and propose a variation of the Fastest Path First scheduling suggested in the paper [45], which is also referred to as the Earliest Delivery Path First in [23] and has the property of eliminating reordering fully in case if all the packets are of the same size. For each arriving packet p the expected delivery time t_{pi} if sent to route i is to be estimated. Then the packet is sent to the path with the minimum value of t_{pi}. We calculate the expected delivery time for each packet according to Algorithm 1.

If several paths share the value of the estimated delivery time for a packet, we choose a path with the earliest expected arrival time of the last packet sent on this path. If the tie still exists, the path with the smallest sequence number of the last packet sent to the path is chosen.

Complexity considerations

The number of operations that the algorithm performs per packet depends only on the number n of disjoint paths available and is constant when n is

Algorithm 1 Multipath data scheduling algorithm

```
if $t^{\text{now}} < t_i^{\text{free}}$ then
    $t_{pi} \leftarrow t_i^{\text{free}} + D_i + S/B_i$ {route $i$ is busy}
    $t_i^{\text{free}} \leftarrow t_i^{\text{free}} + S/B_i$
else
    $t_{pi} \leftarrow t^{\text{now}} + D_i + S/B_i$ {route $i$ is free}
    $t_i^{\text{free}} \leftarrow t^{\text{now}} + S/B_i$
end if
$t_i^{\text{free}}$ keeps the information about the availability of the path $i$,
$t^{\text{now}}$ - current wall-clock time.
```

fixed. Spacial complexity is linear in the flight size - the number of packets which have been sent but not yet acknowledged. One integer space per packet is used to store packet-to-path assignment in the one-dimensional array and is released after successful arrival of the packet.

3.1.2 mHIP Linux Implementation

To accomplish our prototype we used a Linux-based implementation of HIP (HIPL [1]). mHIP adds support for (i) maintaining multiple security associations for the pair of communicating peers, (ii) periodic path probing, and (iii) multipath scheduler in userspace implementation IPSec to perform multipath routing.

Connection establishment

As we have reviewed in Section 2.1.1, any HIP-enabled host triggers a Base Exchange (BEX) procedure for each newly created transport layer connection if no security associations existed previously between the two peers. The result of such a handshake is an IPsec tunnel between the two communicating peers and all subsequent transport layer traffic is to be carried inside this tunnel. HIP uses a *3-way update mechanism* to notify the responder about any changes in network interfaces.

Consider that host A and host B established an IPSec tunnel using HIP BEX. If host A changes its IP address, e.g., due to moving to a different network, it will trigger a 3-way handshake with host B and replace the existing IPSec tunnel with a new one that will correspond to a new IP address of A. Our multipath extension uses the procedure described above with one important modification: if host A adds a new interface it will update host B using a similar mechanism, but instead of replacing the existing IPSec tunnel with a new one, hosts A and B will add it as a second tunnel such that two will co-exist.

If host A is multihomed, there will be multiple IPSec tunnels between A and B. This is not a part of standard IPSec and we have found it rather non-trivial to introduce the required changes to Linux kernel IPSec implementation. Instead, we have used a userspace IPSec to implement support for multiple security associations (or IPSec tunnels) between two peers, periodic tunnel probing (to measure delay and available bandwidth) and a multipath data scheduler. This userspace IPSec is a part of the HIPL project and has similar functionality as its kernel version. Figure 3.2 illustrates the relationship and interaction between the application, the probing mechanism residing in the HIP daemon and the multipath scheduler residing in the IPSec daemon.

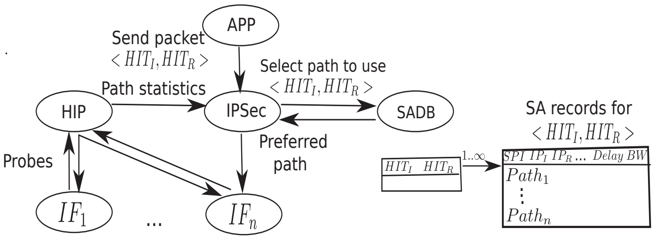

Figure 3.2: Scheduler operation.

Path probing

Once hosts A and B have established security associations, mHIP starts sending periodic probing packets to measure available bandwidth and delay (in our prototype, measurements are done in both directions from A to B and vice versa to address the paths asymmetry problem). The probing packets are carried inside HIP signalling packets, which we call *periodic heartbeat probes*.

We calculate smoothed round trip time (RTT) to measure delays and *packet pair* [137] to estimate the available bandwidth. At the receiver we measure the *dispersion* or inter-arrival time for the packet pair immediately after receiving it from the particular path. It then sends a response to the sender using the same path from which the request has arrived. The response packet, a special type of HIP packet, contains: (i) packet pair dispersion, (ii) the second packet's transmission timestamp, and (iii) processing delay occurring on the receiver. While the information in (i) allows the sender to estimate the available path capacity using packet pair bandwidth estimation; (ii) and (iii) are used to infer RTT. These measurement results are then used by the sender to update the particular security association with fresh path statistics.

Data scheduling

The multipath data scheduler begins its operation when a user datagram becomes available from, e.g., the transport layer. The mHIP scheduler resides inside the userspace IPSec. It intercepts the calls to a function that retrieves the security associations by a given HIT pair. And thus, the scheduler uses the source and destination IPv6 (named Host Identity Tag, or HIT) addresses to identify available security associations. For each obtained security association the scheduler calculates the estimated delivery time as if the packet would have been forwarded into the given tunnel, and makes the scheduling decision according to the multipath scheduling algorithm described above. The forwarding decision then corresponds to a *security association (SA)*, i, with the minimal estimated arrival time. Once the best SA record is selected, the IPSec layer queues the packet to the appropriate interface.

Security considerations

Designed for secure identification of two peers in the network, HIP provides several fundamental security properties: first, it allows to identify communicating peers, including peer authentication during mobility events; second, it reduces the possibilities for denial of service (DoS) attacks using a cryptographic puzzle mechanism; and third, it diminishes the possibility for a number of attacks on the IP layer such as IP address spoofing. All these properties have a special importance for securing multipath TCP communication. To name just a few aspects, HIP protects the transport layer from injections of TCP reset packets and also helps to secure mobility updates. We note that none of these features are seamlessly available to other multipath protocols such as MPTCP.

3.1.3 Some Experimental Results

First we simulated the operation of the multipath scheduler in the ideal network to place a limit on the best performance which is possible to achieve by using multiple paths simultaneously. The simulations were performed using the publicly available ns-2 simulator [3]. We evaluated the performance of our algorithm implemented on two simple network topologies with two and three available paths between one source-destination pair. In

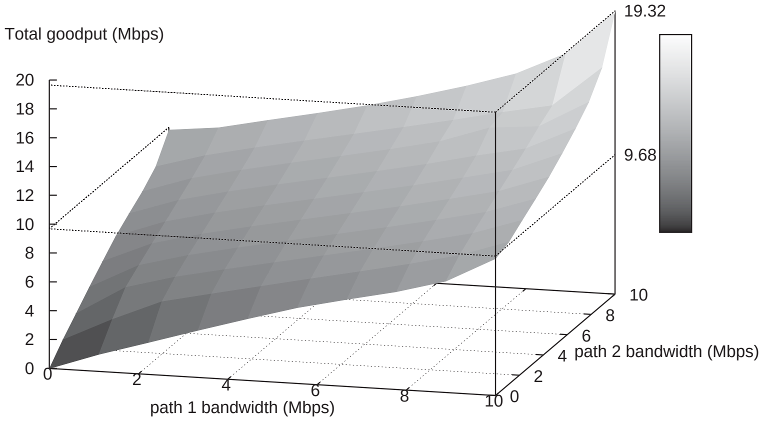

Figure 3.3: Total goodput of the ideal two-path system.

the *ideal system* characteristics of the paths are stable with no packet losses or cross-traffic interruptions.

We measure *goodput*, which is the application-level throughput, the amount of data per unit of time delivered by the network from source to destination, excluding protocol overhead and retransmitted packets.

As shown in Figure 3.3 the increase of the goodput is almost linear provided by the increase of the total bandwidth. The resulting goodput values are compared to the sums of the goodputs of two corresponding unipath systems. The result of the comparison confirms that the two-path system produces about 99% of the sum of the two single path bandwidths.

In the ideal system no reordering occurs at the receiver during the simulation period. This observation confirms applicability and effectiveness of the proposed multipath scheduling algorithm.

Experiments in real networks

In what follows we present the results of our first experiments in the real networks. It should be noticed that neither these results nor mHIP implementation details have yet been published. The author chose to present these results here in order to confirm feasibility and viability of the proposed multipath design ideas and outline future work directions.

We conducted a series of experiments in real networks with the mHIP prototype implemented on both communicated hosts, one of which is multihomed. First we connected the hosts using two Ethernet links (wired links), then in the next experiment one of the paths was replaced by a Wifi connection (a single radio interface connected to a 802.11b Wifi access

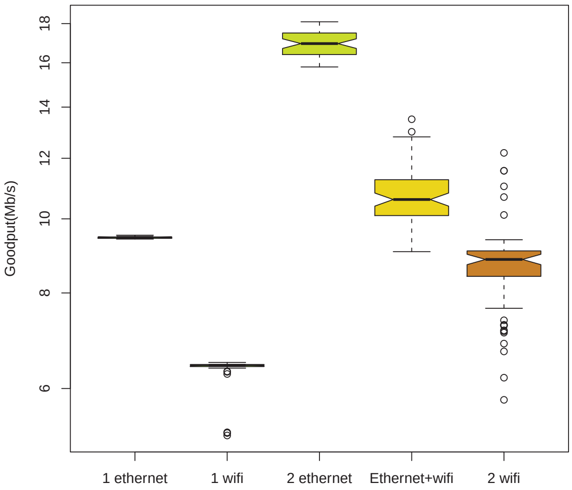

Figure 3.4: Goodput comparison results for three scenarios with different path types.

point), and finally we conducted a similar experiment with the two Wifi paths (both connected to different 802.11n wireless routers (operated on non-overlapping channels)). Figure 3.4 presents the resulting goodput for five experimental scenarios: with all the traffic sent to a unipath network with a single Ethernet or Wifi link, and to the multipath networks with different combinations of these link types. All the experiments were repeated 50 times, and the median values and deviations are illustrated in the figure.

It was not surprising that the scenario with two stable wired links showed the best performance, and the multipath network doubled the goodput of the unipath systems, confirming correctness and applicability of our scheme for wired networks. In the second scenario, where the combination of wireless and wired paths was used, the multipath scheme also demonstrated a noticeable performance improvement: the median for multipath was higher than the maximum for any of the single paths, though the desired performance increase was not stable and varied significantly for different trials. And of course, the most tricky part was the last experiment with two wireless links, where we observed significant variation of the results, with the average goodput still exceeding the goodputs of the single paths.

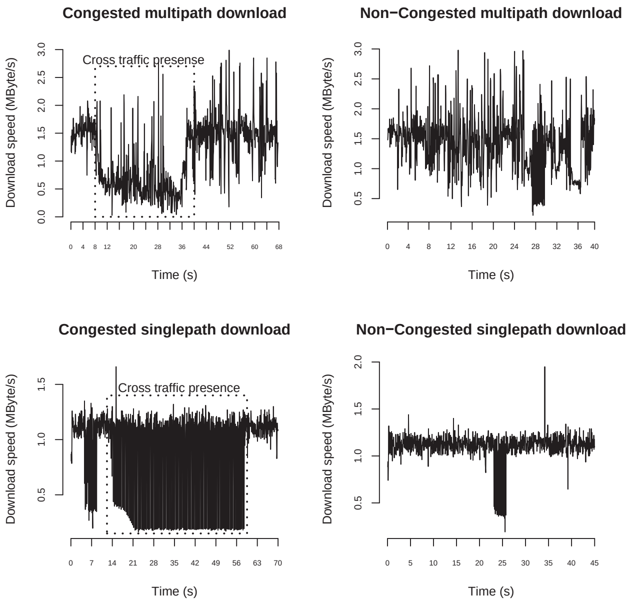

Figure 3.5: Multipath data scheduling with mHIP within the two-path network in the presence of cross-traffic.

Experiments with cross-traffic and losses

The following experiments required the presence of a third node in the network, which generated cross-traffic at a certain bit rate in the direction from the multihomed node towards the cross-traffic receiver. This setup allowed us to congest, in a controlled way, the downlink of one of the paths.

The results illustrated in Figure 3.5, demonstrate how the multipath scheduler redistributes the flows between paths when cross-traffic adds additional delay on one of the paths. When one of the paths experiences a cross-traffic presence, the scheduler sends the data flow towards the less congested path. Moreover, to decrease the packet reordering and utilize the links more efficiently, in the situations with severe congestion and losses the scheduler may temporarily stop sending the packets into the congested path until the path characteristics become stable.

From the results of our initial experimentation we conclude that the proposed design may not be applicable as an out-of-the-box solution for wireless network scenarios. Additional modifications and parameter tuning is required in order to achieve the maximum performance of the multipath system. However, even with current implementation a multihomed node can benefit, when the goal is to increase reliability and robustness to a congestion situation in the wireless network scenario. Later we applied multiple adjustments to our prototype including a receiver buffer with loss differentiation, congestion control and avoidance and more. We continue working on the performance improvement of the mHIP implementation.

3.2 Improving TCP-friendliness and Fairness for mHIP

In Publication II we propose a TCP-friendly congestion control scheme for mHIP secure multipath scheduling solution. We enable two-level control over aggressiveness of the multipath flows to prevent stealing bandwidth from the traditional transport connections in the shared bottleneck. We demonstrate how to achieve a desired level of friendliness at the expense of inessential performance degradation. A series of simulations verifies that mHIP meets the criteria of TCP-compatibility, TCP-equivalence and TCP-equal share, preserving friendliness to UDP and other mHIP traffic. Additionally we show that the proposed congestion control scheme improves TCP-fairness of mHIP.

3.2.1 Two-level Congestion Control for mHIP

We want our mHIP connections to coexist with other traffic providing opportunities for all to progress satisfactorily. To limit aggressiveness of the flow growth we propose the following two-level congestion control scheme: a combination of a per-path AIMD and TCP global stream congestion control on top of it. Additionally, we introduce a sender-side buffer to provide better control on the packet sequence in congestion situations.

The proposed congestion control scheme is illustrated in Figure 3.6. The global congestion controller coordinates the work of the individual per-path controllers and balances the traffic load between the paths according to their available capacity. If the *cwnd* capacity of the quickest path is exceeded, the path with the next minimum estimated arrival time is chosen.

An important property of the proposed scheme is that per-path controllers are connected so that the aggregated congestion window is a simple sum of per-flow congestion windows. The same rule applies to the threshold values. Connecting per-path congestion control parameters in such a way we guarantee the resulting multipath bundle behaves as a single TCP if all packets are sent to the same path.

It should be noted that the congestion control parameters of the global

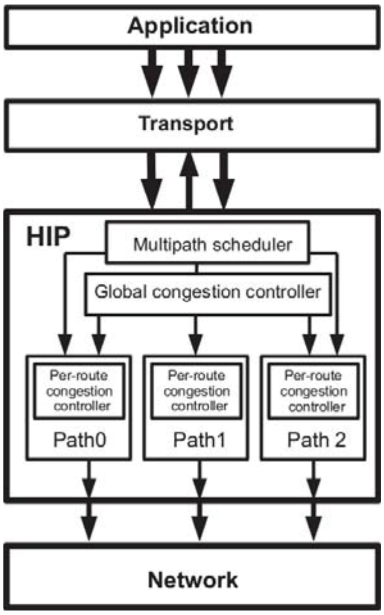
Application
Transport
HIP
Multipath scheduler
Global congestion controller
Per-route congestion controller
Path0
Per-route congestion controller
Path1
Per-route congestion controller
Path 2
Network

Figure 3.6: Two-level multipath congestion control for mHIP.

TCP flow differs from the standard TCP New Reno only in the way the congestion control window grows and decreases (AIMD parameters): the increase of the global *cwnd* is now dictated by the cumulative increase of the per-flow congestion windows and the reaction on losses (*dupack action*) has changed so that the global *cwnd* is not divided by half, but only the window corresponding to the path from which the packet was lost, is decreasing.

3.2.2 Balancing between Aggressiveness and Responsiveness

Our first simulation experiments (described in Publication II) with the proposed congestion control scheme demonstrated the fact that the mHIP flow behaves too leniently when it competes against a standard TCP in a shared link, and is not able to occupy available bandwidth effectively.

We discovered that the problem is in the inability of the mHIP receiver to differentiate between the reordering signals and actual packet losses. In response to the congestion the mHIP scheduler halves the congestion window of the corresponding path, reducing the aggressiveness of the traffic flow. This precaution could be too strict in case when the missing sequence numbers are not lost but just slightly delayed in competition with the external flows.

To cope with the problem we propose to increase the *dupthresh* value

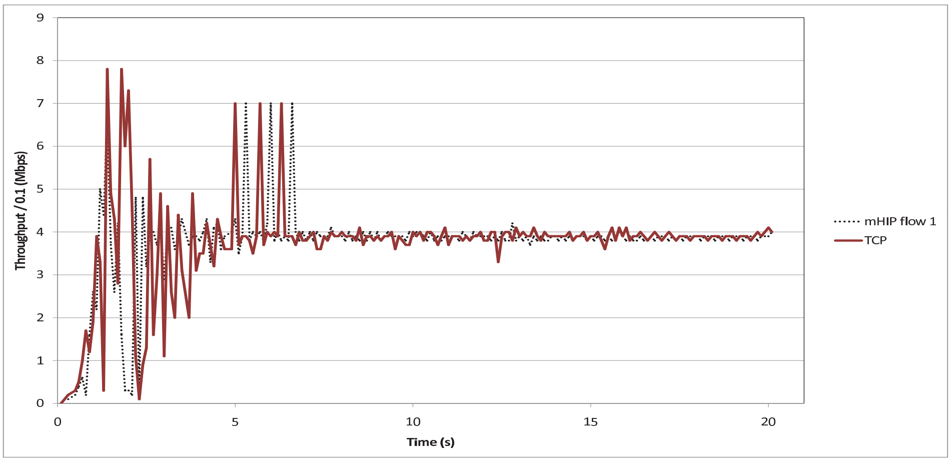

Figure 3.7: mHIP flow 1 coexists in a friendly manner with a TCP New Reno flow.

and introduce a new time variable ADDR (allowable delay due to reordering), which stores how much time has elapsed since the congestion situation in some path was reported. If the missing sequence number arrives successfully during ADDR, *cwnd* and *ssthresh* of the path should be returned to the values prior to the congestion notification.

Additionally, we locate a sufficiently large buffer at the receiver and include the SACK [69] together with SMART options [81] into our multipath congestion control scheme.

3.2.3 Experimental Validation

Below we provide the final experimental validation of the effectiveness of the proposed modifications to the mHIP congestion control.

TCP-friendliness

Figure 3.7 illustrates how both mHIP and TCP flows competing for a 8 Mbps bandwidth of a shared link are able to achieve comparable average throughputs of $T(mHIP1) = 3.80$ Mbps and $T(TCP) = 3.71$ Mbps with the friendliness factor $FF = \frac{T(mHIP1)}{T(TCP)} = 1.02$. The competition demonstrated high variation about the average during a short stabilization phase. This unfairness is rather moderate and can be tolerated as long as the flows quickly achieve stability and later coexist in a friendly manner.

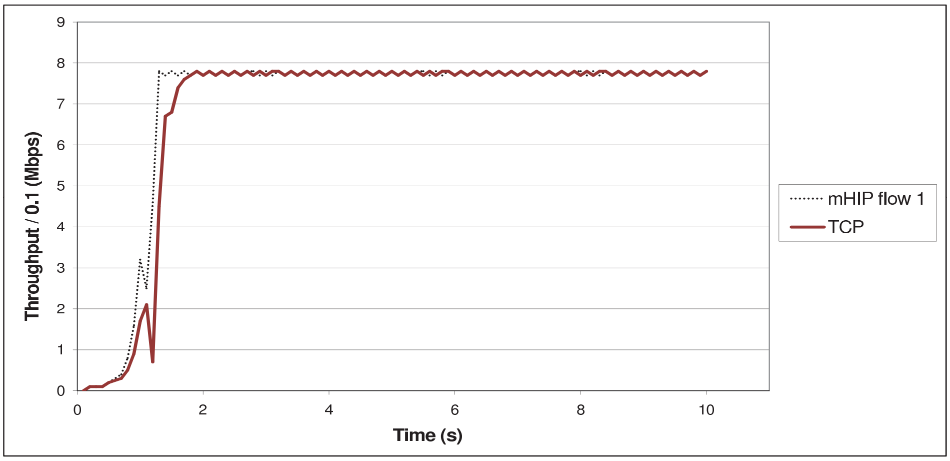

Figure 3.8: Testing TCP-compatibility and equivalence of mHIP.

TCP-compatibility and TCP-equivalence

Figure 3.8 shows that the mHIP flow occupies no more available bandwidth than a TCP flow sent to the same path making it TCP-compatible. Moreover, mHIP achieves the same average flow throughput of 7.8 Mbps as TCP in the steady state and thus meets the criteria of TCP-equivalence.

TCP-fairness in the shared bottlenecks

A flow is *TCP-fair* if its arrival rate does not exceed the rate of a conformant TCP connection in the same circumstances. Put another way, a TCP-fair flow sharing a bottleneck link with N other flows should receive less than or equal to $1/(N+1)$ of the available bandwidth.

Multiple experiments with various path characteristics confirmed that mHIP flows inside one TCP connection share available bandwidth mostly fairly and is still friendly to the external TCP flow. The observed friendliness factor lies within the interval $[0.95, 1.03]$. A typical example of such a bandwidth distribution is shown in Figure 3.9. The mHIP bundle behaves almost as a standard TCP when all of its flows occasionally meet in one link. This result confirms that after we improved the congestion control scheme and limited the increase of the global TCP congestion window, our mHIP solution also meets the TCP-fairness criterion.

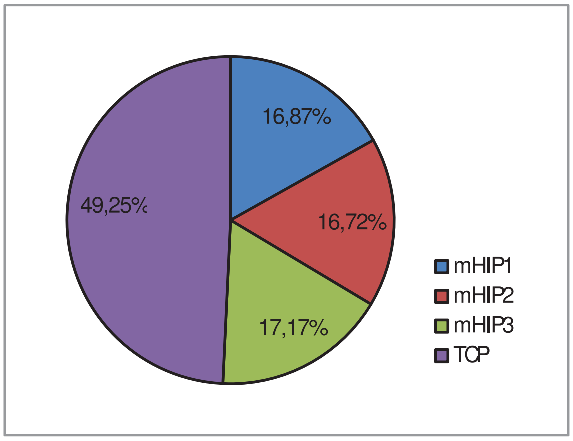

Figure 3.9: Three mHIP flows from one connection compete against one TCP NewReno for the bottleneck bandwidth.

The cost of friendliness

We achieved the desired level of TCP-friendliness for our multipath HIP solution and would like to evaluate the cost in terms of performance degradation paid for this improvement.

We compared the total throughput TT of the traffic flow controlled by multipath HIP with and without the two-level congestion control scheme applied. A number of experiments with different network conditions showed that the desired TCP-friendliness can be achieved at the cost of about 15-20% performance degradation.

3.3 Game-theoretic Approach in Multipath Network Sharing

In Publication III we apply game-theoretic analytical modelling to solve the fair resource allocation problem. We investigate how the multihomed multipath-enabled hosts coexist in the same network, and attempt to answer the following questions: Do the multiple multipath-enabled users with selfish objectives each exploiting similar scheduling techniques share the multipath network fairly? Is such a network sharing optimal or does is need to be improved by applying some global congestion controllers on a higher level?

3.3.1 Multiuser Multipath Network Routing Game

First we formulate the problem as a non-cooperative static routing game and construct a Wardrop equilibrium model with splittable traffic. The amount of flow to route through the network is a variable whose value is set optimally, simultaneously with the routes, as a function of network characteristics and the users demand. Minimization of the end-to-end traffic delay for each user is the criterion of optimality.

The problem is modelled as the game $\Gamma = \langle n, m, w, f \rangle$, where n users send their TCP traffic through m parallel routes from the source S to destination D as shown in Figure 3.10.

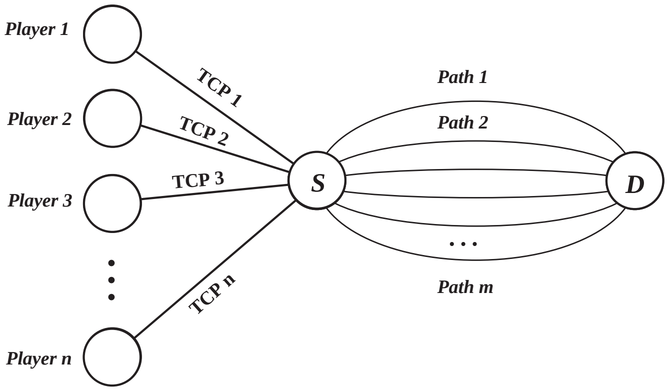

Figure 3.10: Multiuser multipath network model.

Each user of the network is multihomed, which gives him the ability to deliver his traffic along multiple paths simultaneously. The global TCP congestion window grows and shrinks according to the TCP New Reno AIMD (additive increase multiplicative decrease) policy. The change in the window size, which occurs when a new acknowledgement message is received by the source from the receiver, represents a step in the decision-making process. At each step a user makes identical decisions on how to split the given amount w_i of his TCP traffic flow among the available paths.

The users act selfishly and choose routes to minimize their maximal traffic delay. A user's i strategy is $x_i = \{x_{ie} \geq 0\}$, where x_{ie} is the traffic amount that he sends on the path e so that $\sum_{e=1}^{m} x_{ie} = w_i$. Then $x = (x_1, \ldots, x_n)$ is the strategy profile. Denote for the original profile x the new profile $(x_{-i}, x'_i) = (x_1, \ldots, x_{i-1}, x'_i, x_{i+1}, \ldots, x_n)$ where user i changes his strategy from x_i to x'_i and all other users keep their strategies the same.

Each path e has some characteristics, which depend on the end-to-end path parameters, such as propagation delay D_e and bottleneck link bandwidth B_e. The total load of the path e is a function $\delta_e(x)$ that is continuous and non-decreasing by x_{ie}. A continuous traffic delay function $f_{ie}(x) = g_{ie}(\delta_e(x))$ is defined for each user i and each route e. It is non-decreasing by the path load and hence by x_{ie}.

Function $PC_i(x)$ defines the individual i-th user's costs. Each user i tries to minimize his individual costs – the maximal traffic delay among the routes that he uses $PC_i(x) = \max\limits_{e:x_{ie}>0} f_{ie}(x)$.

Social costs depend on the users' traffic volume $w = (w_1, \ldots, w_n)$, characteristics of the paths and users' strategies. Here social costs are the total traffic delay on the paths of the network [39]:

$$SC(x) = \sum_{i=1}^{n}\sum_{e=1}^{m} x_{ie} f_{ie}(x).$$

3.3.2 Routing Game with Traffic Delay Function $1 - e^{-\alpha_e \delta_e(x)}$

The amount of time needed to traverse a single path of a network is typically load-dependent, that is, the traffic latency in a path increases as it becomes more congested. Based on a series of simulations of TCP traffic with variable path characteristics, we choose a traffic delay function $f_{ie}(\delta) = 1 - e^{-\alpha_{ie}\delta}$ to approximate the dependency between the end-to-end delay of the TCP traffic controlled by New Reno loss-recovery [37] in combination with drop-tail queue management, and the total path load δ. TCP regulates the load by relying on the packet loss and reduces the rate in response to that. When the path load is large, packet loss on the path is large too, so it prevents an infinite growth of the delay.

In the model with the traffic delay function $f_{ie}(x) = 1 - e^{-\alpha_{ie}\delta_e(x)}$, where $\delta_e(x) = \sum\limits_{i=1}^{n} x_{ie}$, Nash and Wardrop equilibria are obviously coincident, because Property 2 (described in Section 2.1.4) holds.

The social costs are $SC(x) = W - \sum\limits_{i=1}^{n}\sum\limits_{e=1}^{m} x_{ie} e^{-\alpha_{ie}\delta_e(x)}$, where $W = \sum_{i=1}^{n} w_i$ – is a total traffic in the network.

We suppose that traffic delay on a path e is the same for each user and equals $f_e(x) = 1 - e^{-\alpha_e\delta_e(x)}$, resulting in $SC(x) = W - \sum_{e=1}^{m} \delta_e(x) e^{-\alpha_e\delta_e(x)}$.

Wardrop Equilibrium

Let a profile x be a user's profile in a Wardrop equilibrium. By definition if $x_{ie} > 0$ then $f_e(x) = \min\limits_{l} f_l(x) = \lambda_i$ and if $x_{ie} = 0$ then $f_e(x) \geq \lambda_i$. Since traffic delay on the path e is equal for all users, for each i, such that $x_{ie} > 0$, $\lambda_i = \lambda$. Delays on the unused routes are equal to zero, that is why in the Nash equilibrium each path must be used by at least one user. Moreover, if for some user i on the path e the traffic load is $x_{ie} = 0$, then traffic delay on this path must not be less than delays on the paths which he uses, i.e. $1 - e^{-\alpha_e \delta_e(x)} \geq \lambda > 0$. It means that there is at least one user k, such that $x_{ke} > 0$, hence the traffic delay on this path is exactly equal to λ. So, in the Wardrop equilibrium traffic delays on each route equal to λ and for all $e \in \{1, \dots, m\}$ holds $\delta_e(x) = -\frac{\ln(1-\lambda)}{\alpha_e}$.

Summing these expressions by e we get

$$W = -\ln(1-\lambda) \sum_{e=1}^{m} \frac{1}{\alpha_e}, \text{ and } \lambda = 1 - e^{-\frac{W}{\sum\limits_{e=1}^{m} \frac{1}{\alpha_e}}}.$$

Substituting λ into the expression for $\delta_e(x)$ we obtain that in a Wardrop equilibrium loads are distributed by routes as follows:

$$\sum_{i=1}^{n} x_{ie} = \delta_e(x) = \frac{W}{\alpha_e \sum_{e=1}^{m} \frac{1}{\alpha_e}} \text{ for each } e \in \{1, \dots, m\},$$

According to the Karush-Kuhn-Tucker theorem, x is a stationary point if for each user i and each link e, such that $x_{ie} > 0$, holds

$$\frac{\partial}{\partial x_{ie}} \left(SC(x) - \sum_{i=1}^{n} \gamma_i \left(\sum_{e=1}^{m} x_{ie} - w_i \right) \right) = 0$$

$$\text{or } e^{-\alpha_e \delta_e(x)} (\alpha_e \delta_e(x) - 1) = \gamma_i.$$

In equilibrium $1 - e^{-\alpha_e \delta_e(x)} = \lambda_i$ for all e, or $\alpha_e \delta_e(x) = -\ln(1-\lambda_i) =$ *const* by e, which satisfies the requirement to be a stationary point, but the question of its social optimality needs to be investigated more.

Social costs depend on the users' traffic volume $w = (w_1, \dots, w_n)$, characteristics of the paths and users' strategies. However, since $e^a \geq 1 + a$ for $a > 0$, we can give a lower estimation $LSC(x)$ for our social costs function:

$$SC(x) \geq LSC(x) = W - \sum_{e=1}^{m} \frac{\delta_e(x)}{1 + \alpha_e \delta_e(x)}.$$

The function $LSC(x)$ is convex, so it has a unique minimum, which is also global. The stationary point for $SC(x)$ is also a stationary point for its lower estimation $LSC(x)$. Thus, the minimum for $LSC(x)$ and a lower estimation for $SC(x)$ is Wardrop equilibrium profile x^{WE}, such that $\delta_e(x^{WE}) = \frac{W}{\alpha_e \sum_{e=1}^{m} \frac{1}{\alpha_e}}$:

$$\begin{aligned} SC(x) \geq LSC(x^{WE}) \quad &= W - \textstyle\sum_{l=1}^{m} \left(\frac{1}{\alpha_l} \frac{W}{\left(\sum_{e=1}^{m} \frac{1}{\alpha_e}\right)\left(1+\alpha_l \frac{W}{\alpha_l \sum_{e=1}^{m} \frac{1}{\alpha_e}}\right)} \right) \\ &= W - \textstyle\sum_{e=1}^{m} \frac{1}{\alpha_e} \frac{W}{\left(\sum_{e=1}^{m} \frac{1}{\alpha_e}\right)\left(1+\frac{W}{\sum_{e=1}^{m} \frac{1}{\alpha_e}}\right)} \\ &= W - \frac{W}{1+\frac{W}{\sum_{e=1}^{m} \frac{1}{\alpha_e}}} = W \left(1 - \frac{1}{1+\frac{W}{\sum_{e=1}^{m} \frac{1}{\alpha_e}}} \right). \end{aligned}$$

The Price of Anarchy

Now we can estimate the Price of Anarchy for the game with parallel paths, defined as a ratio of equilibrium social costs and the optimal social costs. Obviously its lower bound is 1, since a Wardrop equilibrium can be an optimal profile. According to the result from the previous subsection we can give an upper estimation for Price of Anarchy as follows:

$$PoA(\Gamma) = \frac{SC(x^{WE})}{SC_{opt}} \leq \left(1 - e^{-\frac{W}{\sum_{e=1}^{m} \frac{1}{\alpha_e}}}\right) / \left(1 - \frac{1}{1 + \frac{W}{\sum_{e=1}^{m} \frac{1}{\alpha_e}}}\right).$$

Denote $\frac{W}{\sum_{e=1}^{m} \frac{1}{\alpha_e}}$ as $C \geq 0$.

Then,

$$PoA \leq (1 - e^{-C})(1 + \frac{1}{C}).$$

This function has one maximum on interval $[0; +\infty)$ and its maximal value is about 1.3, leading to the total latency of each user in the Wardrop equilibrium not being higher than a small constant times that of a system optimum.

3.3.3 Experimental Example

Consider the multipath scheduling problem described earlier in Publication I and in Section 3.1.1 of this thesis. Traffic sent by a user is presented as a sequence of data packets each of size S located at the sender. m available paths connect the sender and the receiver, each of which could consist of a number of consecutively connected links, with the following end-to-end path characteristics: D_e - delay in the path e; B_e - bottleneck bandwidth of the path e. According to the proposed model if a packet is sent to a busy channel it will arrive to the receiver at the time $t_e^{\text{free}} + S/B_e$, where t_e^{free} indicates the time when this path becomes free after delivering previously sent packets. If N packets are sent to the same route, the next packet sent will be delayed by $N * S/B_e$. Here N is roughly the number of packets in progress, or the current load of the path δ_e.

According to the multipath scheduling algorithm proposed as an optimal solution to the multipath scheduling problem, the optimal strategy of the user is to distribute packets among paths according to their capacities.

Now we apply the model to our multipath multiuser routing game with the only addition that we allow more than one user to use the same network. We set the parameter of the route $\alpha_e = S/B_e$. Then the path load of each of our users profile in Wardrop equilibrium $\delta_e(x) = \frac{W*B_e}{\sum_{j=1}^{m} B_j}$. The loads are distributed by the routes as $\sum_{i=1}^{n} x_{ie} = \delta_e(x)$ for each path $e \in \{1, \ldots, m\}$. The equilibrium strategy for each user in the multiuser game is to distribute the traffic load among the paths according to their capacities and it coincides with the optimal strategy proposed for a single user in Publication I. The result confirms the correctness of our choice of traffic delay function for approximation of TCP-controlled flows.

We simulated a multipath multiuser game using ns-2 network simulator [3] in order to evaluate the price of anarchy for a chosen setup. Six multipath TCP agents are attached to the source of the 3-path network connecting the source and destination nodes. The paths' bandwidths were chosen as follows: 8 Mbps (megabit per second), 4 Mbps and 4 Mbps (16 Mbps network total) with the corresponding propagation delays: 60ms, 60ms and 20ms, which provide diversity in the path parameters. Each user sends 15 Mbytes of individual TCP traffic (90 Mbytes total). The resulting traffic delays for each of the six users correspond to their personal costs in equilibrium and are distributed as follows: 48.84, 47.02, 47.09, 48.23, 46.91, 45.08 s.

We compare the total equilibrium social costs $SC(eq) = 48.84$ s to the theoretical optimum, which corresponds to the minimum possible delay of 90 Mbytes traffic in such a network $SC(opt) = 45$ s. And the price of anarchy $PA = SC(eq)/SC(opt) = 1.082 < 1.3$, which is consistent with the theoretical results presented above.

3.4 Improving Scalability of Real-time Data Transmission

In Publication IV we design a scalable multicast architecture for potentially large overlay networks. Our techniques address suboptimality of the adaptive hybrid error correction (AHEC) [148] scheme in the multicast scenarios. A hierarchical multi-stage multicast tree topology is constructed in order to improve performance of AHEC and guarantee QoS for the multicast clients.

Consider a general network $\mathcal{G}$. One node S of the network is the *sender*. A set R of nodes are the *receiver* nodes. Each link in $\mathcal{G}$ is characterized by its round trip time (RTT) and packet loss rate (PLR). We integrate the parameters of the link into a single number that we call the link's *cost*, which roughly defines the redundancy level required to serve the link. Based on the costs, we construct the tree $\mathcal{T} = (V, E)$ of shortest paths from the main sender S to all receivers in R.

3.4.1 Control Nodes and Regions

We divide the multicast tree $\mathcal{T}$ into subtrees, called *regions*. The root of the region is called the *control node* C_j, which serves the individual redundancy and retransmission requirements of the receivers within the region.

Region *size* s_j is the longest distance in terms of costs from the control node to any receiver within this region. It should not exceed the *maximum cost per region*: $s_j \leq c_{max}$, which defines the ability of control nodes to serve the receivers of the assigned regions.

Control node functionality

A control node receives data from the source, stores the current in-flight data in a buffer, decodes the content and forwards data directly to the end receivers. The control node also retransmits missing data segments upon request from individual receivers within its region. Furthermore, it is possible to isolate *bad* receivers and limit their influence on the nodes outside of their regions. One control node can control either a large number of receivers connected to the node through good quality (i.e. low cost) links, or fewer receivers connected through links with worse quality (i.e. higher cost), or a combination of different types of receivers.

The overhead of the insertion of control nodes is in fact negligible. The router assigned to be a control node stores only a small amount of data packets in flight to be able to serve retransmission requests from the receivers. Obviously modern routers possess a sufficient amount of memory to store several seconds the data from the multicast data stream. The decoding times are also small in comparison to the end-to-end paths RTT values.

3.4.2 Control Nodes Assignment Algorithm

The algorithm finds optimal placements for control nodes needed to serve all the receivers minimizing the total number of control nodes.

Let us define a receiver R_i to be *served* if it has a control node C_j within the distance c_{max} from it. The control node is called *critical* if it lies at the maximum distance from R_i.

The algorithm is scanning through the tree from the leaves up to the root starting from the most remote receiver. It is consequently checking whether the condition $|R_i n_k| \leq c_{max}$ is satisfied for the current node n_k. If for the following node $|R_i n_{k+1}| > c_{max}$ (or if we reached the tree root), then n_k is the critical node for the receiver R_i and n_k is assigned to be a control node C_j. The whole subtree with the root in C_j is assigned to be the $j-th$ region and is removed from further consideration. The most remote receiver is to be found in the rest of the tree and the scanning repeats. The procedure stops when each receiver of the original multicast tree is served. A pseudo code of this algorithm is provided in Algorithm 2.

Algorithm 2 Control nodes assignment algorithm

```
for all receivers in the tree do
  find distances from the root to all the receivers
  find $R_i$ - the most remote receiver
  $n_k$ - current node
  while $n_k \neq root$ do
    go up the tree $k = k + 1$
    find the critical node for $R_i$ and assign it to be a control node $C_j$.
    remove subtree with the root in $C_j$ from the tree
  end while
  $j = j + 1$
end for
```

Next we prove correctness of the proposed algorithm. Let l be the most remote receiver. Let c be the first control node found by the algorithm. In any feasible solution, the subtree with the root in c must contain a control node c'. Otherwise l is not served. We claim that without loss of generality,

c' can be shifted to c. Indeed, suppose that shifting c' to c makes some leaf l' unserved. This means that $|l'c| > c_{max}$. But we know that $|lc| \leq c_{max}$, and for the main source s we have $|l's| = |l'c| + |cs| > |lc| + |cs| = |ls|$, which contradicts the assumption that l is the furthest receiver. We conclude that since on each iteration the algorithm starts from the most remote receiver l, the distance to any other node in the subtree with a root in c will be less than $|cl|$.

The algorithm is fast with the running time $O(n \log m)$, where $m \log m$ is required to sort m receivers. Note that in the Internet-like topologies for which the algorithm is designed, the number of receivers is regularly less than half of the total number of tree nodes [32].

3.4.3 Redundancy Optimization

Inspired by the previous work [151] we continue working on optimizing the amount of redundancy information required by AHEC to better serve the needs of particular multimedia applications. The total needed redundancy for each receiver depends on the number of retransmission rounds, which in turn has a direct connection to the end-to-end RTT and PLR values. In the multicast scenario the amount of redundancy is dictated by the worst receiver in the group. By introducing control nodes between the source and end receivers we shorten the RTT needed for retransmissions and isolate the links with high loss probability so that they influence only the performance of the corresponding region, but not the whole multicast tree.

Redundancy Information

In what follows we calculate RI_{HEC}, the redundancy information amount required by the HEC-PR scheme, according to the formulas provided in Section 2.2.3. In HEC-PR the number of source data packets in one encoding block $k = 1$, and the architecture acts as a pure ARQ-based scheme. The redundant packets during all retransmissions are always the copies of the source data packets.

Again, we are minimizing the total needed redundancy information by optimizing the number of retransmission rounds needed to provide P_{target} without D_{target} violation:

$$RI_{HEC}^{opt} = \arg\min RI_{HEC}, \text{ s.t. } 1 \leq \tilde{N_{rr}} \leq N_{rr}$$

Shannon limit

The Shannon's coding theorem [96] defines the theoretical maximum information transfer rate of the binary erasure channel with the error probability P to be equal to $1-P$. Therefore, for reliable data transmission the amount of redundancy information should be at least $RI_{lim} = \frac{P}{1-P}$.

Obviously, in practice we need slightly more than this theoretical value. Our goal is to minimize the redundancy information amount used in the error correction scheme.

Proximity

Optimization brings the redundancy information amount closer to the theoretical lower bound. In order to compare the performance of the AHEC scheme for different network settings we introduce the following metric: *Proximity ϵ is the difference between the amount of redundancy information introduced into the network by the AHEC scheme for the current system setup and the desired optimal value of the redundancy required for that system according to Shannon bound: $\epsilon = RI_{HEC} - RI_{lim}$.*

Proximity takes non-negative values only and shows how close the AHEC scheme approximates the optimum.

Multi-hop redundancy

In a traditional error correction scenario the required amount of redundancy must be carried along the whole end-to-end virtual link, and each underlying physical link segment also has to carry this amount. In our scheme control nodes break paths from the source to receivers into several segments. The average end-to-end path redundancy is $RI_{e2e}^{avg} = RI_{HEC}$. Assuming a path is split into N segments, the average multi-hop path redundancy is
$RI_{mh}^{avg} = \frac{\sum_{i=1}^{N} RI_{HEC,i}}{N}$.

Note that this holds for calculation of both RI_{HEC} and RI_{lim} values. And when we need to estimate proximity for the multi-hop case, we in fact compare RI_{HEC}^{avg} and RI_{lim}^{avg} over the undrerlined segments.

Cost metric

Cost metric is a crucial factor for the optimization process since it directly relates the network characteristics to the mathematical calculations. We experimented with several cost metrics to test the effectiveness of our multi-stage multicast scheme: 1) costs equal to RTT values, 2) costs equal to

PLR values or 3) costs equal to $RTT \cdot PLR$, and studied the sensitivity of the amount of redundancy to each of them. Each metric demonstrated the influence on the results of the optimization procedure. And the most noticeable dependency was observed between the first metric (costs equal RTT) and the amount of redundancy in the HEC-PR scheme.

Figure 3.11 illustrates how the amount of the HEC-PR redundancy reacts to the changes in RTT value for a fixed PLR range. The initial network parameters were set as described in [151]. If we take a point with the fixed $PLR = 0.1$ and the corresponding optimal redundancy $RI_{lim} = \frac{PLR}{1-PLR} = 0.1(1)$, then setting $RTT = 20ms$ brings RI_{HEC} quite close to the RI_{lim}, with the proximity $\epsilon \leq 0.0001$. Further reduction of RTT will not significantly improve the RI_{HEC} value, but increase the complexity of network management.

We discovered that for each bounded range of PLR we can find an appropriate threshold RTT value, such that the corresponding HEC scheme results in the redundancy close to the optimal with a certain fixed proximity. This showed us how to choose the right maximum cost per region value c_{max}. First we choose the desired proximity ϵ_{target}. Then for the fixed PLR we start increasing the RTT value within the allowable range and calculate the corresponding proximity until it reaches the desired ϵ_{target}, and choose c_{max} equal to the minimum RTT for which $\epsilon \leq \epsilon_{target}$.

Problem Statement

The redundancy optimization problem can be described as follows.

Given a fixed network topology with known PLR and RTT values of all links and a given multicast scenario (e.g. HDTV) with known data rate, target delay D_{target} and residual error rate P_{target}, find the threshold c_{max} in order to bring the redundancy closer to the Shannon limit within ϵ_{target}.

We are not aiming to reach the optimal Shannon bound for each case, but to find the appropriate limiting cost values, which guarantee the redundancy is close to the optimum with the given proximity ϵ_{target}.

Optimization Procedure

Step 1. For the given tree we identify the worst receiver with respect to the *cost* metric, calculate the required amount of redundancy RI_{HEC} and the resulting proximity ϵ. If the proximity is greater than the desired ϵ_{target} we introduce control nodes into the tree. For the fixed PLR_{e2e} of the worst receiver and variable RTT values we find the threshold, after which proximity exceeds ϵ_{target}, which gives us the c_{max} value. We apply

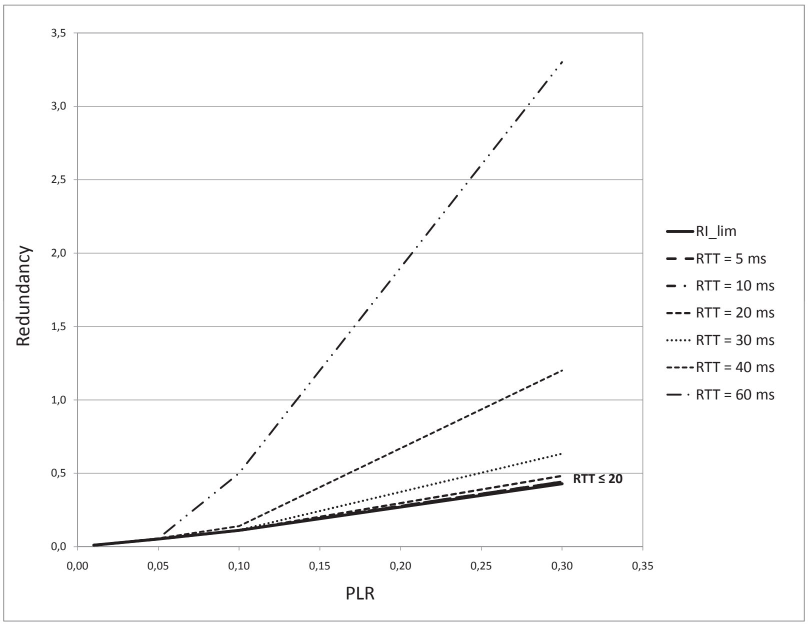

Figure 3.11: *RI* sensitivity to the changes in RTT.

the control node assignment algorithm (Algorithm 2) with respect to the optimum c_{max} value. The control nodes break the end-to-end paths to the receivers into several segments. We recalculate the multi-hop redu ndancy RI_{HEC} for the worst path as an average among the segments together with the corresponding proximity. If the new proximity ϵ' is closer to the desired ϵ_{target}, we proceed to the next iteration. Otherwise the procedure stops, we conclude that for the given setup it is not possible to achieve the desired ϵ_{target} and report the best proximity we have achieved, the corresponding number of control nodes and recommend their optimal placement.

Step 2. For the second step of the optimization procedure we reconstruct the tree in the following way: the control nodes of the regions obtained on the previous iteration are now replacing the whole region and the other nodes assigned to the region (if they are not control nodes for the other regions) are relaxed from further consideration. We also adjust the D_{target} value by subtracting the maximum cost among the regions. This operation reflects the fact that some part of D_{target} was consumed inside the lower regions, which in the worst case corresponds to the cost of the worst region. Next, we repeat all the steps as described for *Step 1*.

After each iteration the tree recursively shrinks and converges to the root. The optimization procedure stops when either the desired proximity

ϵ_{target} is achieved or if after a certain iteration the proximity is not improving, which means it is not possible to optimize the redundancy further.

3.4.4 Empirical Evaluation

In order to benchmark the efficiency of the proposed redundancy optimization scheme we conducted a series of simulations with the initial parameter set as specified in Table 3.1. The chosen low values for P_{target} and D_{target} are required for such demanding applications as DVB services or gaming. Multicast tree topologies adhere to the power-low distributions typical for the multicast trees extracted from the real Internet topologies as described in [32]. The redundancy optimization procedure goes in several iterations.

Table 3.1: Simulation parameters

P_{target}	D_{target}	ϵ_{target}	RTT_{link}	PLR_{link}
10^{-6}	$200ms$	10^{-5}	$10 \ldots 50$ ms	$10^{-3} \ldots 10^{-2}$

In Publication IV we provided a numerical example with a 10-node tree, which elucidates how the proposed optimization works step by step. We illustrated the three iterations of the redundancy optimization procedure, after which the desired proximity ϵ_{target} was successfully reached.

Next we conducted a series of experiments with different tree sizes. For each tree size the experiment was repeated one hundred times. We applied the redundancy optimization procedure and measured the resulting proximity. Figure 3.12 illustrates average relative improvement of proximity for each of the tree sizes. For each tree size, the proximity was significantly improved, which means that the total amount of required redundancy information was reduced, and in fact in some cases it achieved the theoretical Shannon bound.

As we mentioned before the overhead of such a scheme is negligible, adding control nodes into the system implies minimal changes in the router functionalities and the total number of control nodes in the tree is not very large. Table 3.2 shows the number of control nodes required for each of tree size considered in the experiments.

Note that the size of the multicast tree is limited by the chosen parameter set. Since the AHEC framework guarantees D_{target} and P_{target} for the application, they limit the total end-to-end delay for the worst receiver, which in our case corresponds to the depth of a multicast tree.

The depth of the multicast tree is a critical parameter, while the width of the tree in terms of the number of branches, could be chosen arbitrarily

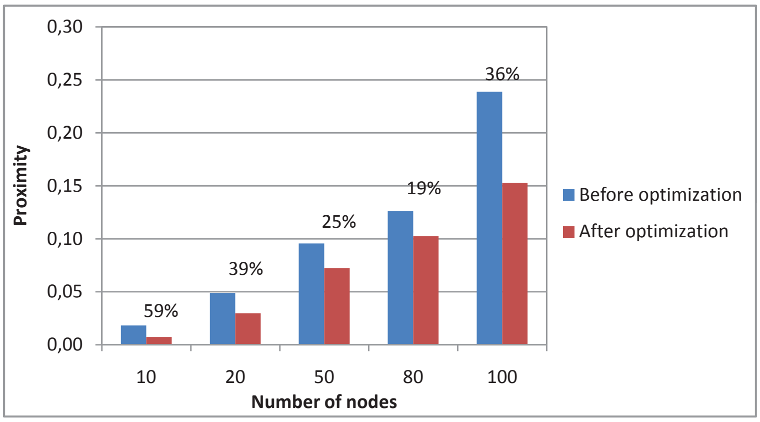

Figure 3.12: The improvement observed in the proximity (estimated in percent from the value before optimization) for different tree sizes.

Table 3.2: Total number of control nodes

Tree size	min	max	average
10	1	3	1.6
20	1	6	4.1
50	2	17	9.4
80	6	21	14.0
100	7	22	15.9

large in general. In our special case depth of a 100-node multicast tree, which was constructed according to the power-law degree distribution, can achieve 15 hops in some cases, and with the maximum $RTT_{link} = 50$ ms it can easily exceed the available $D_{target} = 200$. This makes the trees with bigger depths initially infeasible. The limitation applies only to the chosen application parameter set. The whole multi-stage multicast architecture could easily be applied to the wider range of application scenarios, and we believe has a potential to provide the desired QoS for much bigger multicast client groups.

3.5 Mediating Multimedia Traffic With Strict Delivery Constraints

In Publication V we propose an evolutionary networking approach that has a potential to lower the required resources for multimedia applications. We are targeting the broadcasting and multicasting real-time transmission scenarios. Additionally, we also consider such new application scenarios as stereoscopic streams where the basic stream can be extended by a secondary one to allow a subset of devices displaying the content. The rest of the devices are still receiving legacy streams.

The proposed solution is based on two ideas. First, we propose to reduce the network load by tailoring error-correction schemes to both their application scope and underlying network topology. Furthermore, we introduce the idea how to exploit parallel networks by insertion of supplementary data into the primary network where the appropriate content is available. It leads to a relief of traffic in parts of the network. Eventually, the amount of saved load can be spent on other services or on a larger number of receivers by using the actual network topology and devices.

To take care of timely delivery and an upper limit for residual errors at the receivers, both approaches make use of an error-correction domain separation [75]. We propose to implement these functions as operating modes of the multi-purpose nodes, which we call *Mediators* following their operating principle of mediating traffic between multiple network segments. Thereby, Mediator nodes are introduced into the network where it is appropriate, and they divide subsequent links into several segments. This way non error-prone links are released from carrying redundant data required by error-prone links as it happens in traditional end-to-end environments.

3.5.1 Problem Statement

The problem area considered in this work is identified as follows. We assume the data distribution structure is established in form of a tree $T = (V, E)$ of the size $|V| = N$. We consider multicasting and broadcasting transmission scenarios for real-time multimedia applications. The real-time traffic imposes a tight upper limit for its delivery time Δ and the residual error loss rate P_{target} at the receiver. To achieve the desired *quality of experience (QoE)* [107] both constraints must be completely satisfied. One of the following error-correction schemes: *forward error correction (FEC)*, *automatic repeat request (ARQ)* or *hybrid error-correction (HEC)* is applied for data protection.

The targeted applications are replenished by the new arising transmission scenario where primary data can be extended by a *supplementary data*, which is sent independently from a different source. We assume that the second stream revalues the primary one. In this case, injecting the supplementary data at suitable locations within the network is a crucial factor to lower the total network load.

The transmitted data consists of two parts: the pure payload traffic and extra sent data to cope with transmission errors. In the course of this work we call the former type *primary* traffic and the latter type *redundancy*. The main objective is to find a mechanism that reduces the amount of traffic whereas the operating conditions of the transmission and the connected applications are not disturbed. Specifically, we propose a way to lower the amount of redundancy since the primary traffic can be reduced only by applying more efficient source-coding mechanisms.

3.5.2 Node Characteristics

Current networks evolve an increasing number of different physical transmission mechanisms as fiber and copper links, wireless LAN or 3G and 4G connections, resulting in highly heterogeneous topologies. In general, network nodes serve as routers dealing with forwarding packets to the right next hop. Multi-purpose nodes are able to operate in different ways: as a *routing relay*, *error-correction relay*, and *supplementary data injector*. Further we call such multi-purpose nodes *Mediators* following their operating principle of mediating traffic between multiple network segments.

Routing Relay

The first and simplest operation mode of Mediators is the *routing relay mode*. In this mode the node acts as a normal network router, simply forwarding the data packets to the right next hop that is closer to the designated destination of the data. In this case, no additional enhancements for the node are required. No additional complexity is introduced. Thus, all the existing network routers are operating in this mode. We assume *routing relay* is the default mode for Mediators.

Error-Correcting Relay

A mediator operating in the *error-correcting relay* mode splits the end-to-end transmission path into multiple segments. It enables a precise and individual application of error-correction schemes to the particular network

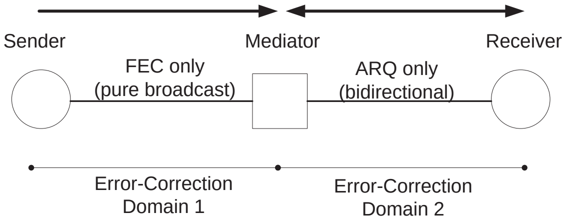

Figure 3.13: Error-Correction Relay.

sections. Thus, an explicit loss domain separation is established. This domain separation frees insusceptible links from traffic introduced by error-prone segments on the same path. The error-correction relay mode works with both pure broadcasting correction schemes as forward error correction (FEC) and bidirectional correction approaches as automatic repeat request (ARQ). A combination of both schemes to the hybrid error-correction (e.g. HEC) is also conceivable, especially in case of time and residual loss restricted transmission conditions. In contrast to the aforementioned mode an error-correction relay requires more resources as CPU power and larger buffers for a reliable application of error protection techniques. The exact overhead depends on the specific error-correction approach applied.

Figure 3.13 presents a schematic application of the two different error-correction schemes for two separated path segments resulting in areas of individual error-correction domains on the path. Obviously, all the ARQ rounds stress only segment 2, whereas segment 1 does not suffer from the extra rounds.

Supplementary Data Gateway

The *supplementary data gateway* mode does not modify the given network characteristics but constitutes a gateway to other networks. A mediator acting in this mode exploits the network topology and the availability of redundant transmission content. The gateway opens multiple additional transmission features.

We distinguish between *horizontal* and *vertical* supplementary data. The *horizontal* supplementary data refers to the traffic containing the same content but sent via other networks. An example for this type is live sports content which could also be sent via satellite or terrestrial propagation besides the IP transmission. We define *vertical* supplementary data as real additional data that revalues the primary data stream. An example of this traffic type could be a program information during an IPTV transmission for hearing-impaired people. Both supplementary data types are not

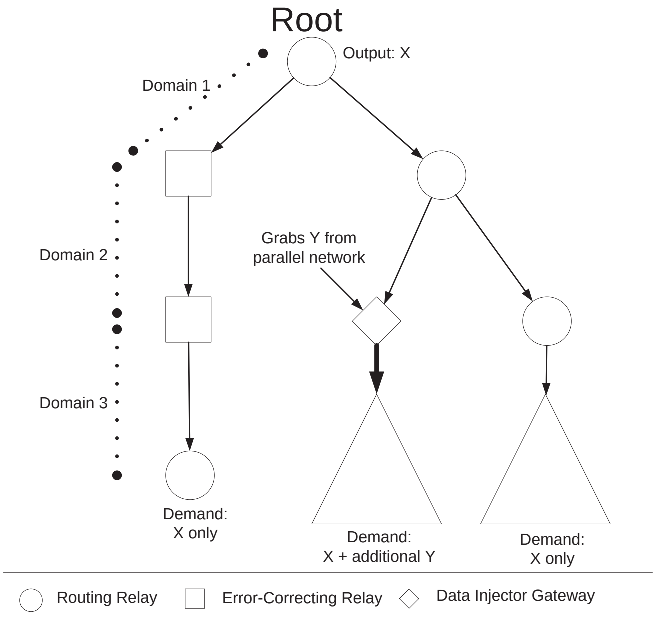

Figure 3.14: Mediators' operation modes.

supposed to be originated by the same source as the primary traffic.

Physical requirements for the *supplementary data gateways* mode are more distinctive as for the preceding modes. To inject supplementary data the node should be able to provide access to other networks via specific interfaces as a DVB-S/T tuner for satellite or terrestrial reception. An important aspect is synchronization of the primary and supplementary data streams. The implementation must provide a way to ensure that both streams fit together perfectly in order to deliver a smooth experience at the receiver.

Mediator nodes are supposed to support one of the last two modes or both. The nodes are aware of their own operation modes and the modes of their neighbors. Figure 3.14 presents a summary of all presented operation modes located within a simple tree network.

3.5.3 Node Location Assignment

Since the Mediator nodes actively influence transmission characteristics, a careful placement of these nodes within the dissemination structure is required. The main objective of the location assignment is to establish an effective error-correction domain separation.

There are two possible scenarios: In the first, the transmission structure is known a priori. In this case all participating nodes are already logically connected by a tree or mesh. In the second case, the transmission structure is not yet known. The dissemination structure can be built with consideration of the possible Mediator locations.

Network Topology-Based Assignment

Potential positions for the Mediator nodes could be identified using the topological characteristics of the network nodes, such as the (weighted) degree, centrality or betweenness of individual network nodes, which we reviewed earlier in Section 2.2.6 of this thesis.

When the set of potential candidates for Mediator node positions have been chosen, the optimization algorithms could be applied. Again, we reviewed the optimization strategies applicable for the Mediator node positioning earlier in Section 2.2.7.

Subjective Assignment

The *Subjective Assignment* depends on a Mediator distribution concept, fully developed by an administrator without any direct reference to the actual network topology. Thus, factors such as hardware capacity, financial considerations, customer's requirements or networking policies within the individual autonomous systems (AS) affect the selection. One must also take into consideration that network providers may deny the application of multi-purpose nodes at certain locations but enforce them, even if the network characteristics do not match with the objective requirements.

Clearly, a combination of multiple assignment strategies to find suitable locations is also possible.

3.5.4 Example Application Scenarios

The proposed Mediator concept can be used with several already established mechanisms and improve their functionality (reliability, resource requirements, etc.). Here we present several example application scenarios, where Mediator nodes could be deployed incrementally within the existing networks in order to optimize their performance.

Content Delivery Networks

Content Delivery Network (CDN) is designed to avoid congested network segments, place the requested content closer to the receiver and improve the content delivery quality, speed and reliability while reducing the network load at the primary source server [120]. The main issues with CDNs are the placement of the surrogate servers, the selection of content and data synchronization. The CDN surrogate server can be seen as a Mediator where 1) data is injected from a parallel network (e.g. DVB-T/DVB-S) to lower the data storage and synchronization effort while releasing parts of the network from carrying data to the receiver and 2) additional error-correction is applied to lower the amount of extra needed bandwidth in the network due to error-prone segments.

Dynamic Adaptive Streaming over HTTP (DASH)

Dynamic Adaptive Streaming over HTTP (DASH) [93] mainly addresses the HTTP-based progressive downloads mechanism, but also attempts to resolve such arising issues as missing bitrate adaptivity or waisted network bandwidth due to user-terminated sessions while further content has already been downloaded [144]. Thus, the server holds a set of differently encoded media chunks and the receiver chooses an appropriate bitrate, thereby changing the quality. To avoid congestion or overload server farms (HTTP caches) are established, that allow highly scalable distribution scenarios. Introducing Mediators into this environment could help to improve the media retrieval process from the source server to the server directly communicating with the receiver: if content is not already available, a reliable and nearly real-time reloading from the source server is possible. Thus, the HTTP caches can be quickly refreshed in multicast mode if required.

Peer-to-Peer Networks

Traditional Peer-to-Peer Networks [19], [88] are overlay networks, built above the physical or logical networks. The main challenge with peer-to-peer network is the high heterogeneity within the set of nodes and connections between the nodes (e.g. DSL, wireless, backbones, etc.).

Recent approaches [109] already incorporate more information from the underlying physical network, and they focus mostly on the financial aspect but not on reliability and network speed. Introducing Mediators into peer-to-peer overlay networks helps to correct transmission errors due to highly error-prone communication links (e.g. IEEE 802.11) by individually protecting these weak links with a better error-correction code, which leads to a lower network utilization when using an additional supplementary data injector. Mediator creation within an overlay network causes a minimal additional setup effort.

Chapter 4

Conclusions and Future Work

In this thesis we proposed several techniques targeted to improve users' experience dealing with the applications utilizing both elastic and real-time types of traffic.

In Publications I-III we developed a multipath-enabled extension for HIP. We proposed a design of an online multipath data scheduling algorithm for HIP, which effectively distributes packets from a TCP connection over available links. It requires modifications only in the HIP daemon at the sender. Legacy IPv4 and IPv6 applications unaware of multiple paths can benefit from it transparently.

Our experiments demonstrated robustness of proposed multipath data scheduling on the HIP layer. In an ideal system with no cross-traffic, overall goodput of the simple multipath system is nearly the sum of link bandwidths. When cross-traffic was introduced to the system, we were able to effectively decrease the number of retransmissions and packet losses. The result was achieved by applying a multipath congestion avoidance scheme, which includes redirection of the traffic to the less congested paths and consequent path probing.

Next we designed and evaluated a TCP-friendly congestion control scheme for mHIP. The traffic splitting algorithm does not explicitly change either the TCP congestion window growing rate or its recovery speed. We showed a way to tune aggressiveness of the multipath data transmission controlled by mHIP without losing its responsiveness in competition with cross-traffic. The proposed two-level congestion control is adjusted to meet the TCP-friendliness and TCP-fairness definitions.

In Publication III we constructed a game-theoretic model to examine the ability of multiple multipath users to share the network with each other in a friendly manner and with the legacy single-path connections while providing the opportunities for all to improve the resulting throughput.

We found an elastic-demand Wardrop equilibrium for splittable traffic, and evaluated the price of anarchy to prove that implementing multipath with an adequate multipath congestion control scheme, selfish users can successfully achieve their personal goals without cooperation, and the resulting unfairness will be rather moderate and could be tolerated.

A multi-stage multicast architecture was proposed in Publication IV to provide scalability of the multimedia data transmission for a wide range of real-time Internet applications. This approach reduces the total network load in the multicast scenarios with heterogeneous receivers by optimizing the amount of redundancy information required for efficient traffic protection with AHEC, keeping it close to the theoretical Shannon limit.

To extend this work further, in Publication V we introduced Mediators, which reduce the redundancy in the system with error-correction by tailoring error-correction schemes to both their application scope and underlying network topology. Furthermore, the Mediators exploit parallel networks for selective supplementary data insertion.

Still many open questions remain for future work. These include a comprehensive evaluation of robustness of the proposed solutions in more realistic dynamic networks scenarios and solving the deployment issues.

The experimental results leave no doubts that the multihomed hosts utilizing ground links can benefit from multipath functionality provided by mHIP. Nevertheless, the current implementation is not optimized for the bandwidth aggregation of the multiple wireless paths. We will continue our attempts in adjusting mHIP implementation for efficient bandwidth aggregation with the paths consisting of WIFI and HSDPA links.

The evolutionary Mediator approach showed a potential to lower the required resources for multimedia applications. We discussed a feasible scheme that enables multi-constraint multimedia applications, such as live IPTV, to use legacy networking algorithms utilizing only one objective value. Therefore, it is possible to find the global optimum routes within the network by bridging legacy algorithms with new arising multimedia applications. We are interested in implementation of the proposed scheme in the real application scenarios with the use of ultra-new broadcast and multicast multimedia technologies.

Internet users expect high-quality experiences dealing with the wide range of requested applications. Affordable and mature technologies are required to fulfil the users' quality expectations. The designers of the future Internet aim at the efficient and flexible distribution platforms that scale to the rising demands. The architectures and techniques described in this thesis take one step into this direction.

References

[1] HIPL website. Available at: `http://infrahip.hiit.fi`. Accessed 18 July 2012.

[2] Hybrid Broadcast Broadband TV. ETSI TS 102 796 - V1.1.1.

[3] The network simulator ns-2. `http://www.isi.edu/nsnam/ns/`, last checked 06/09/2012.

[4] H. Adiseshu, G. Parulkar, and G. Varghese. A reliable and scalable striping protocol. In *Proc. of ACM SIGCOMM*, pages 131–141, 1996.

[5] J. Afzal, T. Stockhammer, T. Gasiba, and W. Xu. Video Streaming over MBMS: A System Design Approach. *Journal of Multimedia*, 1(5):25–35, 2006.

[6] V. Aggarwal, A. Feldmann, and C. Scheideler. Can ISPs and P2P users cooperate for improved performance? *SIGCOMM Comput. Commun. Rev.*, 37(3):29–40, July 2007.

[7] A. Akella, R. Karp, C. Papadimitrou, S. Seshan, and S. Shenker. Selfish behavior and stability of the Internet: a game-theoretic analysis of TCP. In *Proc. of SIGCOMM'02*.

[8] O. Alparslan, N. Akar, and E. Karasan. AIMD-based online MPLS traffic engineering for TCP flows via distributed multi-path routing. *Annales des Télécommunications*, 59(11-12):1353–1371, 2004.

[9] O. Alparslan, N. Akar, and E. Karasan. Combined Use of Prioritized AIMD and Flow-Based Traffic Splitting for Robust TCP Load Balancing. In *Proc. of QofIS'04*, pages 124–133, 2004.

[10] E. Altman, T. Boulogne, R. El-Azouzi, T. Jiménez, and L. Wynter. A survey on networking games in telecommunications. *Comput. Oper. Res.*, 33(2):286–311, Feb. 2006.

[11] T. Aura, M. Roe, and A. Mohammed. Experiences with host-to-host IPsec. In *Proc. of Security Protocols, 13th International Workshop*, Apr. 2005.

[12] I. Aydin, J. Iyengar, P. Conrad, C.-C. Shen, and P. Amer. Evaluating TCP-friendliness in light of Concurrent Multipath Transfer. *Comput. Netw.*, 56(7):1876–1892, May 2012.

[13] S. Barre, C. Paasch, and O. Bonaventure. Multipath TCP: From theory to practice. In *Proc. of IFIP Networking, Valencia*, May 2011.

[14] S. Bhandarkar and A. L. N. Reddy. TCP-DCR: Making TCP robust to non-congestion events. In *Proc. of NETWORKING'04*, pages 712–724, 2004.

[15] E. Blanton and M. Allman. On making TCP more robust to packet reordering. *ACM SIGCOMM Comput. Commun. Rev.*, 32:20–30, Jan. 2002.

[16] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13:422–426, July 1970.

[17] T. Bonald and L. Massoulié. Impact of fairness on Internet performance. In *Proc. of ACM Sigmetrics*, pages 82–91, 2000.

[18] R. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby. Performance enhancing proxies intended to mitigate link-related degradations. RFC 3135, 2001.

[19] J. Buford, H. Yu, and E. K. Lua. *P2P Networking and Applications.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2008.

[20] R. Buyya, M. Pathan, and A. Vakali. *Content Delivery Networks.* Lecture Notes in Electrical Engineering. Springer, 2008.

[21] Z. Cao, Z. Wang, and E. Zegura. Performance of hashing-based schemes for Internet load balancing. In *Proc. of INFOCOM 2000*, volume 1, pages 332–341. IEEE, 2000.

[22] M. Cha, P. Rodriguez, J. Crowcroft, S. B. Moon, and X. Amatriain. Watching television over an IP network. In *Proc. of the Internet Measurement Conference*, pages 71–84, 2008.

[23] K. Chebrolu, B. Raman, and R. R. Rao. A network layer approach to enable TCP over multiple interfaces. *Wirel. Netw.*, 11(5):637–650, 2005.

[24] J. Chesterfield, R. Chakravorty, I. Pratt, S. Banerjee, and P. Rodriguez. Exploiting diversity to enhance multimedia streaming over cellular links. In *Proc. IEEE INFOCOM*, 2005.

[25] D.-M. Chiu and R. Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Comput. Netw. ISDN Syst.*, 17(1):1–14, June 1989.

[26] J. V. Chuiko and V. V. Mazalov. Nash equilibrium in splittable traffic routing problem. In *Proc. of MTE2008, Japan*.

[27] Cisco Systems. Cisco Visual Networking Index: Forecast and Methodology, 2010-2015, June 2011. `http://http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360.pdf`.

[28] D. Cocker. Multiple address service for transport (MAST). In *Proc. of SAINT 2004, Japan*, page 4. IEEE Computer Society, January.

[29] S. Deering. Host Extensions for IP Multicasting RFC1112, 1999.

[30] S. E. Deering and D. R. Cheriton. Multicast Routing in Datagram Internetworks and Extended LANs. *ACM Transactions on Computer Systems*, 8:85–110, 1990.

[31] C. Diot, B. Neil, L. Bryan, and K. D. Balensiefen. Deployment issues for the IP multicast service and architecture. *IEEE Network*, 14:78–88, 2000.

[32] D. Dolev, O. Mokryn, and Y. Shavitt. On multicast trees: structure and size estimation. *IEEE/ACM Trans. Netw.*, 14(3):557–567, 2006.

[33] Q. Du and X. Zhang. Adaptive Low-Complexity Erasure-Correcting Code-Based Protocols for QoS-Driven Mobile Multicast Services. In *Proc. of the Second International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks*, 2005.

[34] E. Elliott. Estimates of error rate for codes on burst-noise channels. *Bell Syst. Tech. Journal*, 42:1977–1997, 1963.

[35] K. Evensen, D. Kaspar, P. Engelstad, A. F. Hansen, C. Griwodz, and P. Halvorsen. A network-layer proxy for bandwidth aggregation and reduction of IP packet reordering. In *Proc. of LCN2009*, pages 585–592.

[36] J. Feng, Z. Ouyang, L. Xu, and B. Ramamurthy. Packet Reordering in High-Speed Networks and Its Impact on High-Speed TCP Variants. In *Proc. of PFLDnet 2007, Fifth International Workshop on Protocols for FAST Long-Distance Networks*, pages 19–24, 2007.

[37] S. Floyd, T. Henderson, and A. Gurtov. The NewReno Modification to TCP's Fast Recovery Algorithm. RFC 3782, IETF, Apr. 2004.

[38] A. Ford, C. Raiciu, M. Handley, S. Barre, and J.Iyengar. Architectural guidelines for multipath TCP development. *Internet Engineering Task Force (IETF)*, (6182):28, March 2011.

[39] M. Gairing, B. Monien, and K. Tiemann. Routing (Un-) Splittable Flow in Games with Player-Specific Linear Latency Functions. In *Proc. of ICALP*, 2006.

[40] R. Garg, A. Kamra, and V. Khurana. A game-theoretic approach towards congestion control in communication networks. *SIGCOMM Comput. Commun. Rev.*, 32(3):47–61, 2002.

[41] T. Gasiba, W. Xu, and T. Stockhammer. Communication Networks Enhanced system design for download and streaming services using Raptor codes. *European Transactions on Telecommunications*, 20(2):159–173, 2009.

[42] E. N. Gilbert. Capacity of a burst-noise channel. *Bell System Technical Journal*, 39:1253–1265, Sept. 1960.

[43] S. Gorinsky. Feedback Modeling in Internet Congestion Control. In *In Proceedings of the NEW2AN*, 2004.

[44] S. Gorinsky, M. Georg, M. Podlesny, and C. Jechlitschek. A Theory of Load Adjustments and its Implications for Congestion Control. *Journal of Internet Engineering, Klidarithmos Press*, 1:82–93, 2007.

[45] R. Greco and G. Galante. Load balancing over multipaths using bandwidth-aware source scheduling. In *Proc. of the 7th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, December 2005.

[46] S. B. Grossman, H. Sivakumar, S. Bailey, and R. L. Grossman. PSockets: The case for application-level network striping for data intensive applications using high speed wide area networks. In *Proceedings of Supercomputing 2000.*

[47] A. Gurtov. TCP performance in the presence of congestion and corruption losses. Master's thesis, 2000.

[48] A. Gurtov. Effect of delays on TCP performance. In *Proceedings of IFIP Personal Wireless Communications*, pages 87–108, 2001.

[49] A. Gurtov. Responding to spurious timeouts in TCP. In *Proc. of IEEE INFOCOM*, 2003.

[50] A. Gurtov. *Host Identity Protocol (HIP): Towards the Secure Mobile Internet*. Wiley and Sons, 2008.

[51] A. Gurtov, M. Komu, and R. Moskowitz. Host Identity Protocol (HIP): Identifier/Locator Split for Host Mobility and Multihoming. *Internet Protocol Journal*, 12(1):27–32, Mar. 2009.

[52] A. Gurtov and J. Korhonen. Measurement and analysis of TCP-friendly rate control for vertical handovers. *ACM MCCR*, 8, 2004.

[53] A. Gurtov, D. Korzun, A. Lukyanenko, and P. Nikander. Hi3: An efficient and secure networking architecture for mobile hosts. *Computer Communications*, 31(10):2457–2467, June 2008.

[54] T. J. Hacker, B. D. Noble, and B. D. Athey. Improving throughput and maintaining fairness using parallel TCP. In *Proc. of IEEE InfoCom*, 2004.

[55] 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.*, 14(6):1260–1271, Dec. 2006.

[56] S. Hassayoun, J. Iyengar, and D. Ros. Dynamic window coupling for multipath congestion control. In *Proc. of the ICNP*, pages 341–352, 2011.

[57] T. Heer and S. Varjonen. Host Identity Protocol Certificates. IETF RFC 6253, March 2011.

[58] T. R. Henderson and A. Gurtov. draft-irtf-hip-experiment-05.txt. IETF RFC 6538, March 2012.

[59] T. R. Henderson, P. Nikander, and M. Komu. Using the Host Identity Protocol with legacy applications. IETF RFC 5338, Sept. 2008.

[60] Y.-S. Ho and S.-H. Kim. Video coding techniques for ubiquitous multimedia services. In *Proceedings of the 1st international conference on Ubiquitous convergence technology*, pages 1–10, Berlin, Heidelberg, 2007. Springer-Verlag.

[61] H. W. Holbrook and D. R. Cheriton. IP multicast channels: Express support for large-scale single-source applications. In *Proceedings of SIGCOMM'99*, pages 65–78.

[62] C. Hopps. Analysis of an equal-cost multi-path algorithm. IETF RFC2992, 2000.

[63] H.-Y. Hsieh and R. Sivakumar. pTCP: An end-to-end transport layer protocol for striped connections. In *Proc. of ICNP*, pages 24–33, 2002.

[64] W. Huffman and V. Pless. *Fundamentals of Error-Correcting Codes.* Cambridge University Press, 2003.

[65] T. Ishida, K. Ueda, and T. Yakoh. Fairness and utilization in multipath network flow optimization. In *Proc. of the IEEE International Conference on Industrial Informatics*, pages 1096–1101, 2006.

[66] M. Ishiyama, M. Kunishi, and F. Teraoka. An analysis of mobility handling in LIN6. In *Proc. of 4th International Symposium on Wireless Personal Multimedia Communications*, 2001.

[67] 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.*, 14(5):951–964, Oct. 2006.

[68] V. Jacobson. Congestion avoidance and control. In *Proc. of Symposium proceedings on Communications architectures and protocols*, SIGCOMM '88, pages 314–329, New York, NY, USA, 1988. ACM.

[69] V. Jacobson and R. T. Braden. TCP extensions for long-delay paths. IETF RFC 1072, 1988.

[70] Z. Jerzak and C. Fetzer. Bloom filter based routing for content-based publish/subscribe. In *Proc. of DEBS '08*, pages 71–81, New York, NY, USA. ACM.

[71] P. Jokela, R. Moskowitz, and P. Nikander. Using the Encapsulating Security Payload (ESP) Transport Format with the Host Identity Protocol (HIP). IETF RFC 5202, Mar. 2008.

[72] P. Jokela, A. Zahemszky, C. Esteve Rothenberg, S. Arianfar, and P. Nikander. LIPSIN: line speed publish/subscribe inter-networking. *SIGCOMM Comput. Commun. Rev.*, 39:195–206, August 2009.

[73] S. Kandula, D. Katabi, B. Davie, and A. Charny. Walking the tightrope: Responsive yet stable traffic engineering. In *Proc. of SIGCOMM*, 2005.

[74] S. Kandula, D. Katabi, S. Sinha, and A. Berger. Dynamic load balancing without packet reordering. *SIGCOMM Comput. Commun. Rev.*, 37(2):51–62, Mar. 2007.

[75] M. Karl and T. Herfet. On the efficient segmentation of network links. In *Proc. of the IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (ISBMSB)*, June 2011.

[76] E. Karpilovsky, L. Breslau, A. Gerber, and S. Sen. Multicast Redux: a First Look at Enterprise Multicast Traffic. In *Proc. of WREN*, pages 55–64, 2009.

[77] D. Kaspar, K. Evensen, P. Engelstad, A. F. Hansen, P. Halvorsen, and C. Griwodz. Enhancing video-on-demand playout over multiple heterogeneous access networks. In *Proc. of CCNC 2010*, pages 47–51, Piscataway, NJ, USA. IEEE Press.

[78] D. Kaspar, K. Evensen, A. F. Hansen, P. Engelstad, P. Halvorsen, and C. Griwodz. An analysis of the heterogeneity and IP packet reordering over multiple wireless networks. In *Proc. of ISCC*, pages 637–642, 2009.

[79] F. Kelly and T. Voice. Stability of end-to-end algorithms for joint routing and rate control. *SIGCOMM Comput. Commun. Rev.*, 35(2):5–12, Apr. 2005.

[80] J. Kempf, J. Arkko, and P. Nikander. Mobile IPv6 Security. *Wirel. Pers. Commun.*, 29(3-4):389–414, June 2004.

[81] S. Keshav and S. P. Morgan. Smart retransmission: Performance with overload and random losses. In *Proceedings of the INFOCOM '97*, pages 1131–1138, Washington, DC, USA. IEEE Computer Society.

[82] P. Key, L. Massoulié, and D. Towsley. Path selection and multipath congestion control. *Commun. ACM*, 54(1):109–116, Jan. 2011.

[83] P. Key and A. Proutiere. Routing games with elastic traffic. *SIGMETRICS Performance Evaluation Review*, 37(2):63–64, 2009.

[84] S. A. Khayam and H. Radha. Markov-based modeling of wireless local area networks. In *Proc. of MSWIM 2003*, pages 100–107, New York, NY, USA. ACM.

[85] K.-H. Kim and K. G. Shin. Improving TCP performance over wireless networks with collaborative multi-homed mobile hosts. In *Proc. of MobiSys 2005*, pages 107–120. ACM.

[86] M. Komu, H. Tschofenig, J. Melen, and A. Keranen. Basic host identity protocol (HIP) extensions for traversal of network address translators. IETF RFC 5770, 2010.

[87] S. Kopparty, S. V. Krishnamurthy, M. Faloutsos, and S. K. Tripathi. Split TCP for mobile ad hoc networks. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM 2002)*, pages 138–142, 2002.

[88] D. Korzun and A. Gurtov. *Structured P2P Systems: Fundamentals of Hierarchical Organization, Routing, Scaling, and Security.* Springer, 2012.

[89] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proc. of 16th Annual Symposium on Theoretical Aspects of Computer Science*, pages 404–413, Trier, Germany, 4–6 Mar. 1999.

[90] J. Laganier and L. Eggert. Host Identity Protocol (HIP) rendezvous extension. IETF RFC 5204, Mar. 2008.

[91] J. Laganier, T. Koponen, and L. Eggert. Host Identity Protocol (HIP) registration extension. IETF RFC 5203, Apr. 2008.

[92] M. Laor and L. Gendel. The effect of packet reordering in a backbone link on application throughput. *Netwrk. Mag. of Global Internetwkg.*, 16(5):28–36, Sept. 2002.

[93] S. Lederer, C. Müller, and C. Timmerer. Dynamic adaptive streaming over http dataset. In *Proc. of MMSys 2012*, pages 89–94, New York, NY, USA. ACM.

[94] K.-C. Leung, V. O. Li, and D. Yang. An overview of packet reordering in Transmission Control Protocol (TCP): Problems, Solutions, and Challenges. *IEEE Transactions on Parallel and Distributed Systems*, 18:522–535, 2007.

[95] B. Li, M. J. Golin, G. F. Italiano, X. Deng, and K. Sohraby. On the optimal placement of web proxies in the Internet. In *Proc. of INFOCOM'99*, pages 1282–1290, 1999.

[96] S. Lin and D. J. Costello. *Error Control Coding, Second Edition.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2004.

[97] H. Liu, V. Ramasubramanian, and E. G. Sirer. Client behavior and feed characteristics of rss, a publish-subscribe system for web micronews. In *IMC'05: Proceedings of the Internet Measurement Conference 2005 on Internet Measurement Conference.* USENIX Association, 2005.

[98] R. Ludwig and R. H. Katz. The Eifel algorithm: making TCP robust against spurious retransmissions. *Computer Communication Review*, 30(1):30–36, 2000.

[99] X. Luo and R. K. C. Chang. Novel approaches to end-to-end packet reordering measurement. In *Proc. of Internet Measurment Conference*, pages 227–238, 2005.

[100] L. Magalhaes and R. H. Kravets. Transport level mechanisms for bandwidth aggregation on mobile hosts. In *Proc. of the Network Protocols, 2001. Ninth International Conference on*, pages 165–171, Nov.

[101] S. Makharia, D. Raychaudhuri, M. Wu, H. Liu, and D. Li. Experimental study on wireless multicast scalability using Merged Hybrid ARQ with staggered adaptive FEC. In *Proc. of the World of Wireless, Mobile and Multimedia Networks, 2008. WoWMoM 2008*, pages 1 –12.

[102] L. Massoulié and P. Key. Schedulable regions and equilibrium cost for multipath flow control: the benefits of coordination. In *Proc. of the 40th Conference on Information Sciences and Systems*, 2006.

[103] M. Mavronicolas and P. G. Spirakis. The price of selfish routing. *Algorithmica*, 48(1):91–126, 2007.

[104] V. Mazalov, B. Monien, F. Schoppmann, and K. Tiemann. Wardrop equilibria and price of stability for bottleneck games with splittable traffic. In *Proc. of 2nd international Workshop on Internet & Network Economics*, 2006.

[105] R. Moskowitz and P. Nikander. Host Identity Protocol architecture. IETF RFC 4423, May 2006.

[106] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson. Experimental Host Identity Protocol (HIP). IETF RFC 5201, Apr. 2008.

[107] M. Mu, E. Cerqueira, F. Boavida, and A. Mauthe. Quality of experience management framework for real-time multimedia applications. *Int. J. Internet Protoc. Technol.*, 4(1):54–64, Mar. 2009.

[108] A. Mull, F. Beer, S. Urquijo, and L. Patino-Studencka. Assisting a global navigation satellite system using a local broadcast network. In *Proc. of BMSB 2011*, pages 1–6. IEEE, June 2011.

[109] M. Mushtaq, U. Abbasi, and T. Ahmed. Network-aware streaming services delivery over ISP-driven P2P networks. In *Proc. of BMSB 2011.*

[110] J. Nagle. On Packet Switches With Infinite Storage. RFC 970, IETF, Dec. 1985.

[111] J. Nash. Non-cooperative games. *Annals of Mathematics*, 54:286–295, Sep., 1951.

[112] T. Nemeth. *OSPF-OMP: Optimized Multi-path Link State Routing.* Honors paper Macalester College. 1999.

[113] P. Nikander, A. Gurtov, and T. R. Henderson. Host Identity Protocol (HIP): Connectivity, Mobility, Multi-Homing, Security, and Privacy over IPv4 and IPv6 Networks. *IEEE Communications Surveys and Tutorials*, 12(2):186–204, 2010.

[114] P. Nikander, T. Henderson, C. Vogt, and J. Arkko. End-host mobility and multihoming with the Host Identity Protocol (HIP). IETF RFC 5206, Apr. 2008.

[115] P. Nikander and J. Laganier. Host Identity Protocol (HIP) domain name system (DNS) extension. IETF RFC 5205, Mar. 2008.

[116] P. Nikander and J. Melen. A bound end-to-end tunnel (BEET) mode for ESP: draft-nikander-esp-beet-mode-09. Work in progress.

[117] E. Nordmark and M. Bagnulo. Shim6: Level 3 multihoming shim protocol for IPv6. IETF RFC 5533, 2009.

[118] T. Opsahl, F. Agneessens, and J. Skvoretz. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks 32*, pages 245–251, 2010.

[119] O'Sullivan. The Internet Multicast Backbone. `http://ntrg.cs.tcd.ie/undergrad/4ba2/multicast/bryan/index.html`. Last accessed Jan 2012.

[120] G. Pallis and A. Vakali. Insight and perspectives for content delivery networks. *Commun. ACM*, 49(1):101–106, Jan. 2006.

[121] P. Paul and S. V. Raghavan. Survey of multicast routing algorithms and protocols. In *Proceedings of the 15th international conference on Computer communication*, ICCC '02, pages 902–926, Washington, DC, USA, 2002. International Council for Computer Communication.

[122] S. Paul, K. K. Sabnani, J. C. Lin, and S. Bhattacharyya. Reliable Multicast Transport Protocol (RMTP). *Selected Areas in Communications, IEEE Journal on*, 3:407 – 421, 1997.

[123] R. Penno, S. Raghunath, and J. Iyengar. LEDBAT Practices and Recommendations for Managing Multiple Concurrent TCP Connections: draft-ietf-ledbat-practices-recommendations-00.txt. Technical report, IETF, Mar. 2009.

[124] S. Pierrel, P. Jokela, and J. M. Melen. Simultaneous Multi-Access extension to the Host Identity Protocol: draft-pierrel-hip-sima-00, June 2006.

[125] N. M. Piratla and A. P. Jayasumana. Reordering of packets due to multipath forwarding - an analysis. In *Proc. of IEEE Int. Conf. on Communications (ICC 2006)*, pages 28–36.

[126] N. M. Piratla, A. P. Jayasumana, A. A. Bare, and T. Banka. Reorder buffer-occupancy density and its application for measurement and evaluation of packet reordering. *Comput. Commun.*, 30(9):1980–1993, June 2007.

[127] J. Qi, P. Neumann, and U. Reimers. Dynamic broadcast. In *Proceedings of 14th ITG Conference on Electronic Media Technology*, pages 1–6, Dortmund, 2011.

[128] A. Qureshi, J. Carlisle, and J. Guttag. Tavarua: Video Streaming with WWAN Striping. In *Proc. of ACM Multimedia 2006*, Santa Barbara, CA, October 2006.

[129] H. Radha and M. Wu. Overlay and peer-to-peer multimedia multicast with network-embedded FEC. In *Proc. of IEEE International Conference on Image Processing (ICIP)*, 2004.

[130] S. Ratnasamy, A. Ermolinskiy, and S. Shenker. Revisiting IP multicast. *SIGCOMM Comput. Commun. Rev.*, 36:15–26, August 2006.

[131] S. Rost and H. Balakrishnan. Rate-aware splitting of aggregate traffic. Technical report, MIT, 2003.

[132] T. Roughgarden, E. Tardos, and va Tardos. How bad is selfish routing? *Journal of the ACM*, 49:236–259, 2001.

[133] S. Shakkottai, E. Altman, and A. Kumar. The Case for Non-Cooperative Multihoming of Users to Access Points in IEEE 802.11 WLANs. In *Proc. of INFOCOM*, 2006.

[134] Y. Shan, I. V. Bajic, S. Kalyanaraman, and J. W. Woods. Overlay multi-hop FEC scheme for video streaming over peer-to-peer networks. *IEEE ICIP*, 2004.

[135] X. Shen, H. Yu, J. Buford, and M. Akon. *Handbook of Peer-to-Peer Networking*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[136] M. Shreedhar and G. Varghese. Efficient fair queueing using deficit round robin. *SIGCOMM Comput. Commun. Rev.*, 25(4):231–242, 1995.

[137] A. Shriram and J. Kaur. Empirical evaluation of techniques for measuring available bandwidth. In *Proc. of INFOCOM'07*, pages 2162–2170, 2007.

[138] V. Singh, T. Karkkainen, J. Ott, S. Ahsan, and L. Eggert. Multipath RTP(MPRTP). Technical report, July 2012. Internet draft, draft-singh-avtcore-mprtp-05, work in progress.

[139] A. C. Snoeren. Adaptive inverse multiplexing for wide-area wireless networks. GLOBECOM, 1999.

[140] R. Srikant. *The Mathematics of Internet Congestion Control*. Systems & Control. Birkhäuser, 2004.

[141] W. R. Stevens. *TCP/IP illustrated (vol. 3): TCP for transactions, HTTP, NNTP, and the Unix domain protocols*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1996.

[142] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. J. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson. Stream Control Transmission Protocol. RFC 2960, IETF, Oct. 2000.

[143] M. Stiemerling, J. Quittek, and L. Eggert. NAT and firewall traversal issues of Host Identity Protocol (HIP) communication. IETF RFC 5207, Apr. 2008.

[144] T. Stockhammer. Dynamic adaptive streaming over HTTP: standards and design principles. In *Proc. of MMSYS 2011*, pages 133–144, New York, NY, USA. ACM.

[145] T. Stockhammer. Robust System and Cross-Layer Design for H.264/AVC-Based Wireless Video Applications. *EURASIP J. Adv. Sig. Proc.*, 2006.

[146] T. H. Szymanski and D. Gilbert. Design of an IPTV Multicast System for Internet Backbone Networks. *Int. J. Digital Multimedia Broadcasting*, 2010.

[147] G. Tan and T. Herfet. Evaluation of the Performance of a Hybrid Error Correction Scheme for DVB Services over IEEE 802.11a. In *Proc. of BMSB 2007*, Orlando, USA, March.

[148] G. Tan and T. Herfet. Hybrid Error Correction Schemes under Strict Delay Constraints. In *Proc. of BMSB 2010.*

[149] G. Tan and T. Herfet. A Novel Adaptive Hybrid Error Correction Scheme for Wireless DVB Services. *IJCNS*, 1(2):187–198, 2008.

[150] G. Tan and T. Herfet. On the architecture of erasure error recovery under strict delay constraints. In *Proc. of the 14th European Wireless Conference*, June 2008.

[151] G. Tan and T. Herfet. Optimization of an RTP level hybrid error correction scheme for DVB services over wireless home networks under strict delay constraints. *Broadcasting, IEEE Transactions*, 53:297, March 2007.

[152] S.-C. Tsao and N. Chiao. Taxonomy and evaluation of TCP-friendly congestion-control schemes on fairness, aggressiveness, and responsiveness. *Network, IEEE*, 21(6):6–15, Nov. 2007.

[153] T. Tung and J. Walrand. Providing QoS for Real-time Applications. In *Proc. of ICCIT 2003*, pages 462–468.

[154] H. Wang, H. Xie, L. Qiu, Y. R. Yang, Y. Zhang, and A. Greenberg. COPE: Traffic engineering in dynamic networks. In *Proceedings of ACM SIGCOMM*, Pisa, Italy, September 2006.

[155] J. Wardrop. Some theoretical aspects of road traffic research. In *Institute of Civil Engineers*, 1952.

[156] D. Wischik, M. Handley, and M. B. Braun. The resource pooling principle. *SIGCOMM Comput. Commun. Rev.*, 38(5):47–52, 2008.

[157] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley. Design, implementation and evaluation of congestion control for multipath TCP. In *Proc. of NSDI 2011*, Berkeley, CA, USA.

[158] M. Wu and H. Radha. Distributed network embedded FEC for real-time multicast applications in multi-hop wireless networks. *Wireless Networks*, 16(5):1447–1458, 2010.

[159] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. G. Liu, and A. Silberschatz. P4P: provider portal for applications. *SIGCOMM Comput. Commun. Rev.*, 38(4):351–362, Aug. 2008.

[160] X. Yuan and Z. Duan. Fair round-robin: A low complexity packet scheduler with proportional and worst-case fairness. *IEEE Transactions on Computers*, 58:365–379, 2009.

[161] A. Zahemszky, P. Jokela, M. Sarela, S. Ruponen, J. Kempf, and P. Nikander. MPSS: Multiprotocol Stateless Switching. In *Proc. of INFOCOM 2010, IEEE*, pages 1–6, May 2010.

[162] A. Zhang, J. Lai, A. Krishnamurthy, L. Peterson, and R. Wang. A transport layer approach for improving end-to-end performance and robustness using redundant paths. In *Proc. of USENIX Annual Technical Conference*, pages 99–112, 2004.

[163] M. Zhang, B. Karp, S. Floyd, and L. Peterson. RR-TCP: A Reordering-Robust TCP with DSACK. In *Proc. of IEEE ICNP*, pages 95–106, 2003.