End-to-End Mechanisms for Rate-Adaptive Multicast Streaming over the Internet

Vom Fachbereich
Elektrotechnik und Informationstechnik
der Technischen Universität Darmstadt
zur Erlangung der Würde eines
Doktor-Ingenieurs (Dr.-Ing.)
genehmigte

Dissertation

von

Dipl.-Ing. Ivica Rimac

geboren am 09.11.1971 in Novi Sad

Referent: Prof. Dr.-Ing. Ralf Steinmetz
Korreferent: Prof. Klara Nahrstedt (Ph.D.)

Tag der Einreichung: 01.11.2004
Tag der mündlichen Prüfung: 21.12.2004

D17
Darmstädter Dissertation

Kurzfassung

Die Anwendung kontinuierlicher Medienströme über paketvermittelte Datennetze gewinnt zunehmend an Popularität. Eine Vielzahl von Radiostationen bietet beispielsweise aktuell ihr Programm mittels Streaming-Technologie über das Internet an. Es ist zu erwarten, dass Video-Streaming in Kürze eine vergleichbare Verbreitung erlangt. Im Gegensatz zu tradionellen Fernseh- und Rundfunksystemen basieren Internet Streaming Lösungen vorwiegend auf Unicast-Kommunikation. Dabei werden die Medienströme auf dem Server repliziert und über dedizierte Punkt-zu-Punkt Verbindungen ausgestrahlt. Somit steigt mit jedem gleichzeitig bedienten Teilnehmer sowohl die Server- als auch die Netzlast. Dies kann sehr leicht zu Skalierungsproblemen führen, insbesondere unter Berücksichtigung der hohen Datenmengen von Video-Strömen. Die Multicast-Kommunikation bietet hier eine praktikable und vielversprechende Alternative zum Unicast an. Da die Daten dabei über einen Punkt-zu-Mehrpunkt Kommunikationskontext ausgestrahlt werden, führt dieses Verfahren zu einer erheblich höheren Skalierbarkeit und Steigerung der Netzeffizienz. Dies gilt insbesondere für das Ausstrahlen von Live-Ereignissen an größere Teilnehmergruppen. Während die Multicast-Technologie auf Netzebene sehr weit vorangeschritten ist, gilt dies nicht für die Transport- und Anwendungsebene.

In der vorliegenden Dissertation zum Thema „End-to-End Mechanisms for Rate-Adaptive Multicast Streaming over the Internet“ werden Verfahren zur Verbesserung des Internet-Streamings basierend auf dem Multicast-Paradigma vorgeschlagen. Die entworfenen Lösungsansätze adressieren Problemstellungen, die sich aus der hohen Skalierbarkeitsanforderung und der Heterogenität der Bandbreitenbedingungen der Empfänger ergeben. Dazu werden die Möglichkeiten von skalierbarem Video ausgenutzt und eine geschichtete Übertragung vorgenommen. Somit wird das Video in mehrere Schichten aufgeteilt und jede dieser Schichten über einen eigenen Multicast-Kanal übertragen. Der modulare Entwurf der entstandenen Lösungen erlaubt deren Integration als Funktionseinheiten auf Transport- bzw. Anwendungsebene.

Die vorgeschlagenen Lösungen in dieser Arbeit stellen sich sowohl aus sender- als auch empfängerseitigen Mechanismen für die Ratenanpassung zusammen. Für die senderseitige Anpassung ist ein Algorithmus entwickelt worden, der unter Berücksichtung der Bandbreitenbeschränkungen der Empfänger das Video in mehrere Schichten optimaler Rate einteilt. Die zu Grunde liegende Optimierungsmetrik ist dahingehend neuartig, als dass sie sowohl Aspekte des Datentransports als auch des Benutzers umfasst. Dazu werden die Bandbreitenbedingungen der Empfänger auf ein nutzenbasiertes Fairness-Maß abgebildet. Um die dazu benötigten Bandbreiteninformationen der Empfänger am Sender zu sammeln, ist ein wahrscheinlichkeitstheoretisches Feedback-Abfrageschema entworfen worden. Diese Lösung ermöglicht die Kontrolle des Feedback-Verkehrs innerhalb statistischer Schranken, was sie flexibel und in hohem Maße skalierbar macht.

Ein Schlüsselaspekt beim Entwurf hochskalierbarer Multicast-Lösungen ist die Verteilung von Rechenlast und die Vermeidung von unnötigen Kontrollnachrichten. Es steht in der Verantwortung jeden Empfängers, Rückschlüsse über seine momentanen Bandbreitenbedingungen zu ziehen, ohne dabei den Sender zu involvieren. Dazu ist in der vorliegenden Arbeit ein dem Stand der Technik entsprechendes Verfahren aufgegriffen und erweitert worden, das auf dem Ermitteln des fairen Bandbreitenanteils mittels mathematischer Modellierung des TCP-Durchsatzes aufbaut. Die Ergebnisse einer umfangreichen Simulationsstudie bestätigen die Anwendbarkeit des erweiterten Verfahrens zur Bestimmung der TCP-fairen Rate eines Mutlicast-Empfängers. Somit kann diese Information unter der Verwendung des in dieser Arbeit vorgeschlagenen Feedback-Schemas an den Sender übermittelt und von diesem zur Ratenoptimierung verwendet werden. Ferner dient der ermittelte Wert auch zur empfängerseitigen Anpassung der Rate. Hierzu wurde eine Timerbasierte Strategie für das Beitreten und Verlassen von Multicast-Gruppen entwickelt, die zu einem sinvollen Kompromiss zwischen benutzergeforderter Laufruhe der Videoübertragung und den Anforderungen des Netzes nach kooperativem Verhalten und Ansprechverhalten bezüglich Stauindikation führt.

Abstract

Continuous media applications over packet-switched networks are becoming more and more popular. Radio stations, for example, already use streaming technology to disseminate their content to users on the Internet, and video streaming services are expected to experience similar popularity. In contrast to traditional television and radio broadcast systems, however, prevalent Internet streaming solutions are based on unicast communication and raise scalability and efficiency issues. Multicast communication provides a promising and viable alternative since it can vastly improve scalability and network efficiency for the aforementioned class of applications. Nevertheless, suitable multicast streaming solutions ready for wide-area deployment are yet to emerge.

In this thesis on *Rate-Adaptive Multicast Streaming* we provide mechanisms for improving multicast video streaming over the Internet. Our solutions address major issues that originate from the requirements for multicast solutions to scale to a large number of receivers and to accommodate the latter's heterogeneity of bandwidth capabilities. Therefore, our work exploits scalable-encoded video and utilizes layered multicast transmission on top of the IP multicast architecture. Choosing a modular design yields flexible techniques that can be integrated as building blocks in different transport and application frameworks.

The proposed hybrid set of solutions includes mechanisms for server-side as well as receiver-driven rate adjustment. For the former purpose, we devise an algorithm that optimally stripes the scalable-encoded data into several media quality enhancing layers considering the distribution of receiver bandwidth capabilities. The underlying optimization metric is novel and incorporates transport as well as user aspects. It provides a mapping from each receiver's bandwidth capability onto a utility-based fairness measure. In order to provide means to the server for discovering the bandwidth capability distribution of the active receivers, we design a feedback scheme based on probabilistic polling. It allows to control the feedback traffic within statistical bounds, thus, making the scheme flexible and scaling to very large receiver populations.

A key aspect in the design of scalable multicast solutions is the distribution of computational tasks and the reduction of control messages. Consequently, each receiver is responsible for inferring its bandwidth capability without involving the server. Therefore, we adopt and improve the state-of-the-art approach for estimating the fair bandwidth share based on TCP throughput modeling. Extensive simulation results prove the applicability of the modified scheme for estimating the TCP-fair rate of a multicast receiver. Thus, this information can be communicated to the source for rate optimization purposes utilizing our feedback scheme. In addition, it serves also for receiver-driven rate adaptation using a timer-based multicast group subscription strategy. Our novel approach yields a reasonable trade-off between the user demand for smooth video transmission and the network requirement of cooperativeness and responsiveness to congestion indication.

Acknowledgments

The compilation of a list of all people who deserve acknowledgment is an endeavor on its own. Most probably this list cannot be exhaustive, and even more likely arranging the list in proper order is impossible.

This thesis is concerned with multicast communication, which grounds on implicit addressing of each recipient utilizing group addressing mechanisms. Hence, we here apply this concept to ensure the correctness of our acknowledgments in an efficient way.

Acknowledgment to whom acknowledgment is due.

To my wife Vesna, my daughter Mia,
and Renato†

Contents

1 Introduction **1**

1.1 A Case for IP Multicast 2

1.2 Problem Description 3

1.2.1 Challenges 5

1.2.2 Assumptions 7

1.3 Scope 8

1.4 Contributions 9

1.5 Structure of the Thesis 11

2 Background **13**

2.1 Overview 13

2.2 Terminology 14

2.3 A Hybrid Rate Adaptation Model 15

2.4 The Network Model 17

2.4.1 The Internet Protocol 17

2.4.2 IP Multicast 18

2.4.3 Data and Message Transport 20

2.5 Media Characteristics 21

2.5.1 Scalable Video 21

2.5.2 Video Utility Functions 23

2.5.3 Video Quality Metrics 25

2.6 Related Work 30

2.6.1 Rate Adaptation 30

2.6.2 Stream Organization and Optimization 34

2.6.3 Feedback Control 36

3 Stream Organization 39
3.1 Motivation 40
3.2 Modeling Receiver Utility Fairness 41
3.2.1 Inter-Session Fairness 41
3.2.2 Receiver Fairness Function 42
3.3 Optimizing Group Rates 44
3.3.1 Intra-Session Performance Metric 45
3.3.2 Optimization Algorithm 46
3.4 Deriving Utility Functions 49
3.5 Experimental Evaluation 51
3.5.1 Impact of the Number of Transmission Rates 53
3.5.2 Comparison with Static Allocation Strategies 54
3.6 Summary 59

4 Feedback Control 61
4.1 Motivation 62
4.2 Feedback Control Scheme 63
4.2.1 Design Overview 64
4.2.2 Control Precision 67
4.2.3 Initialization 69
4.2.4 Dynamic Adjustment 71
4.3 Approximation of Bandwidth Distributions 72
4.3.1 Methodology 74
4.3.2 Results 77
4.4 Experimental Evaluation 78
4.4.1 Experimental Setup 78
4.4.2 Discussion of Results 80
4.5 Summary 82

5 Equation-Based Fair Share Estimation 83
5.1 Motivation 84
5.2 Fairness Criterion 85
5.3 Background: TCP-Throughput Model 86
5.3.1 Loss Measurement 88
5.3.2 Loss Fraction versus Loss Event Rate 92
5.4 Applicability Evaluation 93

5.4.1 Overview 94
5.4.2 Simulation Configuration 96
5.4.3 Artificial Channel 97
5.4.4 Bandwidth-Limited Bottleneck 100
5.4.5 Virtual Packets 104
5.4.6 Impact on Multi-rate Multicast 106
5.5 Loss Measurement Modifications 107
5.5.1 Model Assumptions 108
5.5.2 Loss Event Impact 108
5.6 Evaluation of the Modified Algorithm 110
5.6.1 Artificial Channel 110
5.6.2 Bandwidth-Limited Bottleneck 113
5.6.3 Discussion 116
5.7 Summary 117

6 Subscription Level Management 119
6.1 Motivation 120
6.2 Subscription Strategy 120
6.2.1 Naive Strategy 121
6.2.2 Join and Leave Timers 122
6.2.3 Lazy Strategy 124
6.3 Experimental Evaluation 126
6.3.1 Overview 126
6.3.2 Metrics 128
6.3.3 Simulation Configuration 129
6.3.4 Artificial Channel 131
6.3.5 Bandwidth-Limited Bottleneck 135
6.4 Summary 138

7 Conclusion and Outlook 141
7.1 Conclusion 141
7.2 Outlook 143

A FGS Video Sequence Parameters 145

B Bandwidth Capability Distributions 147
B.1 Uniform Distribution 147
B.2 Normal Distribution 148
B.3 Multi-modal Distribution 149
B.4 Weibull Distribution 150
B.5 Measurement-Based Distribution 151
B.5.1 Paxson Measurement 152
B.5.2 PingER Measurement 153

C Adaptive Rate Control Framework 155
C.1 Design Overview 155
C.2 Equation-based Layered Multicast (ELM) 155

Author's Publications 159

Curriculum Vitae 163

Bibliography 167

Acronyms 183

Nomenclature 187

Chapter 1

Introduction

Communication plays an important role in society and life. Radio and television as well as telecommunications and computer networks allow for the delivery and exchange of information. In particular computer communication networks are becoming increasingly important in providing and maintaining complex information systems that empower our society. Since the Internet evolved to be the platform of choice for digital communications, a trend towards service convergence can be witnessed. For example, access to the Internet over cellular phones, telephony over the Internet, and Internet radio broadcast are already state-of-the-art services. Although new technologies provide the end user with high-bandwidth access, computer and telecommunications networks are only insufficiently prepared for the dissemination of video in a point-to-multipoint manner similar to that of television systems.

In the course of this thesis, mechanisms for improving multicast streaming services over the Internet are investigated and designed. We identify a set of challenging issues originating from the characteristics of streaming media applications and the heterogeneity, unpredictability, and dynamics of the communication channels. They arise from the absence of global Quality of Service (QoS) in the Internet to provide soft real-time guarantees required for streaming media [SN04]. We tackle the issues by developing end-to-end solutions that build on the availability of scalable codecs and utilize layered transmission on top of native multicast forwarding capabilities of the Internet Protocol (IP) multicast architecture.

1.1 A Case for IP Multicast

For many decades traditional television broadcast systems have been providing efficient and highly scalable dissemination services for audiovisual information to very large groups of users. The content is broadcasted to the interested users using dedicated unidirectional channels over different media, namely, terrestrial, cable, satellite. Network resources are dedicated and user equipment is standardized, so that the provided QoS is generally very high. However, resource usage in this centralized and closed infrastructure is rather expensive and economically only reasonable for highly popular content. Spontaneous streaming to a widely distributed set of only hundreds or thousands of end devices is hardly possible and reasonable.

In contrast to traditional television broadcast systems, the Internet provides an open and low-cost data communication platform. While for a long time it was mainly used for non-real-time applications such as email, file transfer, and World Wide Web (WWW), in the last few years real-time applications such as audio and video streaming expand in popularity. The fraction of streaming traffic is expected to continue increasing since developments in network technology enable residential users to transmit and receive high-volume content. Local radio stations, for example, already use streaming technology to disseminate their content to users distributed all over the world. However, in contrast to traditional television and radio broadcast systems, prevalent IP streaming solutions are based on the point-to-point paradigm. Data packets are replicating at the server for each individual communication channel, which raises scalability issues with respect to server and network load.

A very popular attempt to tackle the scalability problem of video dissemination over packet-switched networks is the use of caching architectures and the deployment of Content Delivery Networks (CDNs) [Gri00, RYHE00, Zin03, CN03, XJ04]. However, the above architectures are particularly tailored to Video-on-Demand (VoD) scenarios and require the deployment of overlay infrastructures with components placed at the edges of the communication networks. The components of the architectures (e.g., caches and CDN servers) are expensive to deploy and maintain. Furthermore, the architectures suffer similar limitations regarding public accessibility compared to those of traditional television systems.

A cheaper alternative is provided through recently proposed application-layer overlay systems. They are based on the peer-to-peer paradigm and have been developed as an alternative to client-server architectures. Instead of requiring infrastructure components placed in the network or at the edges, overlay nodes are end hosts and overlay links are end-to-end transport-level unicast connections. The functions and services provided by these overlay systems include content distribution and application-level multicast [CRSZ01, Cha03, XCRK03, CLN04]. However, application-layer overlays have several limitations. In contrast to their physical network counterparts, overlay nodes and links are less reliable. This might lead to frequently changing overlay topologies implying control and management overhead. Furthermore, the usage of point-to-point connections significantly reduces efficiency of network resource usage in the case of simultaneous transmission of the same content to a group of users. The latter holds in particular for streaming of live events of medium to high popularity.

Obviously, an enabling technology for the aforementioned application class is native multipoint communication. A corresponding network service, namely IP multicast, has been proposed by Steve Deering already in 1989 [Dee89]. Although it has been available through the experimental MBone for a number of years, it has seen slow commercial deployment in the Internet. One of the main reasons that stalled its widespread use is attributed to the original IP Any-Source Multicast (ASM) service model, which has been developed for supporting a vast class of applications. The resulting complexity makes the architecture unstable, which has been noticed by major carriers [DLL$^+$00]. The current set of applications driving multicast deployment, however, is typically one-to-many, including video streaming as considered in the context of this thesis. Thus, the simpler Source-Specific Multicast (SSM) service model [DGS$^+$03] has been accepted by the Internet Engineering Task Force (IETF) as an alternative tailored for one-to-many applications. It is a promising service model that recently attracted attention and is expected to further accelerate the commercial deployment of multicast to become a mature network service [DLL$^+$00].

1.2 Problem Description

Multicast applications have a much wider range of requirements than unicast applications. It has been recognized that a single, generic solution for the transport and control of multicast traffic is not possible. Hence, when designing mechanisms on top of IP multicast, the specific needs of applications have to be considered. This holds particularly for multicast streaming applications, where challenging issues arise from the mismatch of the requirements of the media and the network service model provided by the Internet.

Traditional video streaming applications rely on open-loop resource control mechanisms; that is, on explicit reservation and allocation of resources for providing the necessary QoS. Although considerable research has been devoted to the hard problem of QoS provision in heterogeneous networks, it has never been successfully solved[1]. Today, the Internet provides only a best-effort service, and it is debatable whether any proactive QoS mechanism will ever be deployed in the global Internet. Nevertheless, even if network QoS might become available, it will probably be realized by reservation schemes based on aggregated flows.

Consequently, different flows will still compete for resources and cooperativeness will remain a fundamental requirement crucial for the stability of the Internet. Streaming solutions have therefore to incorporate mechanisms to prevent unfairness to other flows. They are expected to adopt the "social" rules and behave cooperatively by reacting to congestion signals and adapting their transmission rates properly. Since in large multicast trees there are usually many different bottlenecks, a multicast streaming session has to accommodate heterogeneous bandwidth demands and constraints at the same time.

The requirement of rate adaptation poses a severe problem to traditional streaming mechanisms. In contrast to bulk data transfer, traditionally encoded video is not elastic and cannot tolerate delaying the transfer of data for rate control purposes. Adaptation is rather performed by truncating data, which leads to sever degradation of quality and user satisfaction.

A more elegant way of introducing adaptivity has only recently been enabled by the invention of scalable coding schemes, some of which are also standardized by the Moving Picture Expert Group (MPEG). Hierarchically (layered) encoded video allows dropping data segments in a controlled way in order to overcome the inelastic characteristics of traditional formats. Moreover, a layered streaming scheme introduces only negligible computational overhead compared to approaches relying on transcoding functionality placed in network nodes [Mau98]. As a side-effect, mechanisms utilizing layered video provide means for handling heterogeneity of network conditions very efficiently.

From the preceding discussion it is obvious that dedicated mechanisms built on top of the IP multicast are necessary in order to meet the specific requirements of

---

[1] An excellent treatment of the problems and solutions coupled to QoS provision in heterogeneous systems is presented in [Sch01].

streaming applications in an efficient manner. Comparably little attention has been paid to the research of these mechanisms due to the unavailability of wide-spread native IP multicast services in commercial networks. Despite recent technical advances including the SSM model and scalable coding schemes, suitable multicast rate adaptation and congestion control schemes ready for wide-area deployment are still yet to emerge. While single-rate protocols have recently made considerable progress [Riz00, WH01], they are insufficiently prepared for streaming to large and heterogeneous receiver sets. On the other hand, multi-rate multicast schemes are much better suited to accommodate these requirements but are currently still in their infancy.

1.2.1 Challenges

The advantages of multi-rate transmission schemes over IP multicast for streaming applications are obvious. Nevertheless, there are several challenges remaining for the development of deployable and stable solutions. These issues are enumerated as follows:

1. **Heterogeneity.** Large multicast trees spread over multiple domains and usually include different bottlenecks. Competing cross traffic in the Internet is normally very heterogeneous and dynamic. Hence, receivers located behind different bottlenecks have different and changing levels of TCP-fair throughput. If each receiver has throughput matching its bandwidth capabilities, obviously the overall satisfaction within the session is maximized. This requires the establishment and maintenance of numerous multicast channels for a single session, which implies substantial management and signaling complexity. For a feasible solution, multi-rate multicast schemes should operate with only a small number of multicast channels.

   The number of allocated groups and their data rates heavily impact the overall satisfaction depending on the distribution of the bandwidth capabilities of active receivers. Since the latter is dynamic and not known a priori, optimizing the stream organization within a session might significantly improve network utilization as well as collective user satisfaction. However, this approach demands for the definition of a reasonable and application-aware optimization metric and an efficient group rate allocation[2] algorithm.

---

[2] We use the term rate allocation throughout the thesis to refer to the end-system process of

2. **Scalability.** Multicast is a very efficient transmission mode for large groups of receivers. As the number of participating receivers increases, scalability problems emerge. The performance of a multicast scheme can be significantly degraded by several factors, and in the worst case it might be even stopped from working.

   Particularly, feedback traffic is a major factor impacting the scalability of a multicast session [Dan89]. Requiring frequent status reporting from each receiver for sender-based rate adaptation might flood nodes and links at the reverse path to the source, respectively the source itself. Avoiding the well-known feedback implosion problem demands for appropriate feedback control mechanisms.

   Similarly to the feedback implosion problem, the network might get flooded with control messages for group management when multiple receivers frequently change their group membership. Finally, the computational complexity of invoked algorithms should be reduced as far as possible for scalability reasons.

3. **TCP-compatibility.** In the context of today's Internet which is dominated by Transmission Control Protocol (TCP) traffic, the IETF recommends flows using other protocols than TCP to behave in a TCP-compatible[3] manner [BCC$^+$98, Flo00].

   Alternative solutions for unicast streaming have been proposed, which rely on an analytical TCP model for estimating the fair share of a flow. Due to the very promising results obtained with the TCP-Friendly Rate Control (TFRC) protocol [HFPW03], the equation-based approach is considered also a promising technique that can be integration into multicast schemes. However, the algorithm originally has been designed in the context of closed-loop unicast control and its applicability to (multi-rate) multicast schemes has not been studied sufficiently.

4. **Smoothness versus reactiveness.** A key feature of IP multicast is its group-oriented communication model, where receiver-driven group member-

---

partitioning the streaming data into a number of multicast channels at the source.

[3] A TCP-compatible flow is required to be responsive to congestion notification, and in steady-state to use no more bandwidth than a conforming TCP flow running under comparable conditions.

ship can be utilized for implicit control of data distribution. Hence, receivers in a multi-rate multicast session control and adjust their own throughput by choosing an appropriate group subscription level.

Responding to congestion indications drastically and probing for available network resources aggressively makes a receiver changing its subscription level frequently. Such a behavior causes pronounced rate fluctuations and users of streaming applications will quite probably become annoyed with the resulting coarse-grained quality changes [Zin03]. Furthermore, it might result in the already mentioned implosion of group membership control traffic. Hence, smoother control strategies are demanded that still provide for a certain degree of responsiveness and aggressiveness in order to achieve a high inter-session fairness.

1.2.2 Assumptions

The aforementioned challenges are addressed in this thesis under the following assumptions:

- **IP multicast and Source-Specific Multicast.** The foundation for our work is formed by the Internet Protocol architecture and its native multicast support (IP multicast). Particularly, Source-Specific Multicast (SSM) [DGS+03] is assumed as the underlying multicast model, which is particularly designed for point-to-multipoint applications. SSM uses only source-based shortest forwarding trees; thus, data of different multicast channels from the same source can be assumed to traverse the same path. Using the group-oriented communication model of IP multicast, receiver-directed group membership can be utilized to implicitly control and fine-tune data distribution within the network. For that purpose the IETF has specified the Internet Group Management Protocol (IGMP) [CDF+02], which allows routers to be configured for explicit tracking of hosts[4] and yet provides very low leave latencies.

- **End-to-end paradigm.** The only assumption made on the network is that routers can forward multicast packets according to the IP multicast protocol [Dee89] as they forward unicast packets according to the IP protocol [Pos81].

---

[4] This feature has been introduced in version 3 of IGMP.

Neither the underlying network topology (including link bandwidth, latency, buffer size, etc.) nor the traffic pattern/matrix (number of flows, data volume, etc.) is known. The network is considered a black box providing a best-effort service and it is assumed to be dominated by TCP or TCP-compatible traffic.

- **Fine-granular scalable video.** We assume video to be encoded in a scalable format with recently developed coding schemes for fine-granular scalable video, such as Fine Granularity Scalability (FGS) in MPEG-4 [PE02] or Progressive Fine Granularity Scalability (PFGS) [WLZ01]. These schemes usually encode the streaming media into a base layer and an enhancement layer. The enhancement information is encoded such that it can be truncated at an arbitrary bit rate. Hence, it can be split and organized into several hierarchical layers each mapped to a separate multicast channel in order to form a layered[5] multicast session. Moreover, the partitioning of the data into allocated layers can be dynamically adjusted so that adaptation of transmission rates is possible.

1.3 Scope

The lack of mature and stable multi-rate multicast rate adaptation mechanisms and protocols is a significant impediment to the deployment of multicast streaming services. Hence, the primary goal of this thesis is to contribute to the development of multi-rate adaptation schemes that are highly scalable and can accommodate a heterogeneous environment. Since we consider the Internet as a black box providing a best-effort service and assume support for native point-to-multipoint packet delivery, the focus of this thesis is on the study and development of end-to-end mechanisms. We particularly consider smooth rate control to account for the characteristics of streaming media and we follow the TCP-compatibility paradigm.

It is out of the scope of the thesis to design and specify a protocol architecture or framework but rather to investigate and develop general algorithms and mechanisms that can be reused as modular building blocks.

---

[5] Note that we focus on hierarchically layered streams. However, the mechanisms might be generally or with some modifications applied to replicated streams or non-hierarchically layered streams as well.

1.4 Contributions

As motivated in the preceding sections, our research is centered around end-to-end mechanisms to enable and improve rate adaptation for multicast streaming over the Internet. We summarize the major contributions of the thesis as follows:

- **Intra-session performance metric.** For capturing the intra-session performance of a multi-rate multicast session, we derived the *receiver utility fairness* function. Using the latter, the multicast source can map from the inferred TCP-compatible rate of a receiver to a corresponding fairness index. While state-of-the-art metrics attempt to consider network-centric aspects only, our metric also enables incorporating application-specific aspects, such as the user satisfaction derived from rate-distortion curves and subjective video assessment. Based on the receiver utility fairness function, the intra-session performance is defined as the average receiver utility fairness index of all receivers. The latter is then used to quantitatively evaluate the benefit of multi-rate multicast schemes and adaptive stream organization.

- **Stream optimization algorithm.** We developed an algorithm for optimized organization of encoded video data into a fixed-size base layer and several adjustable enhancement layers. The objective of existing approaches is either the optimization of the aggregate bandwidth, the minimization of the aggregate distortion, or the maximization of the bandwidth-proportional inter-receiver fairness index. In contrast, our objective is the maximization of an intra-session fairness index based on the *receiver utility fairness* metric. The developed algorithm is generic and can easily be adapted to different video characteristics. To quantitatively evaluate the performance of the adaptive and optimized rate allocation approach, we conducted simulations using our algorithm and rate-distortion curves of MPEG-4 FGS-encoded video sequences from literature.

- **Scalable feedback control.** The developed optimization algorithm requires knowledge about the distribution of receivers' bandwidth capabilities. Since this knowledge has to be gathered from the receivers, we developed a feedback suppression scheme based on probabilistic sampling in order to avoid the problem of feedback implosion. In a first approach, we use the sample values directly to calculate the layering scheme and quantitatively study the impact of the sample size on the performance of the optimization algorithm. In a second approach, we derive a statistical model for estimating the bandwidth capability distribution based on the collected samples. To evaluate the model-based approach and compare it with our first approach, a set of simulations for several theoretical and measured bandwidth distributions have been conducted.

- **TCP-compatible rate estimation.** Receivers are required to estimate their bandwidth capabilities for both choosing a reasonable subset of layers (subscription level), and feeding back the estimate to the sender for stream optimization purposes. Mechanisms utilizing a well-known equation-based approach for calculating a TCP-fair rate have been extensively evaluated for unicast streaming. The gathered results indicate that this technique provides a promising basis for smooth rate adaptation. Only recently existing work on multicast schemes adopted the model-based approach originally designed for closed-loop unicast control. Our experimental results, however, show that naively adopting the algorithm may not provide the expected behavior. Analysis and extensive simulations led to the development of an improved algorithm for equation-based estimation of the TCP-compatible rate of a multicast receiver.

- **Group subscription management.** Performing group subscription decisions simply based on the actual calculated TCP-fair estimate might lead to frequent join and leave decisions due to the inherent variations of the estimated value. We developed a mechanism based on dynamic timers that allows for smooth subscription decisions in order to reduce oscillations otherwise caused by frequent join and leave actions. The parameters of the mechanism are tunable such that its responsiveness to congestion indications respectively its aggressiveness regarding the allocation of available network resources can be adjusted independently. The behavior and performance of our mechanism are discussed and compared to that of a very prominent existing mechanism by means of network simulations.

Figure 1.1 illustrates the elements of a multicast streaming scenario and our major contributions.

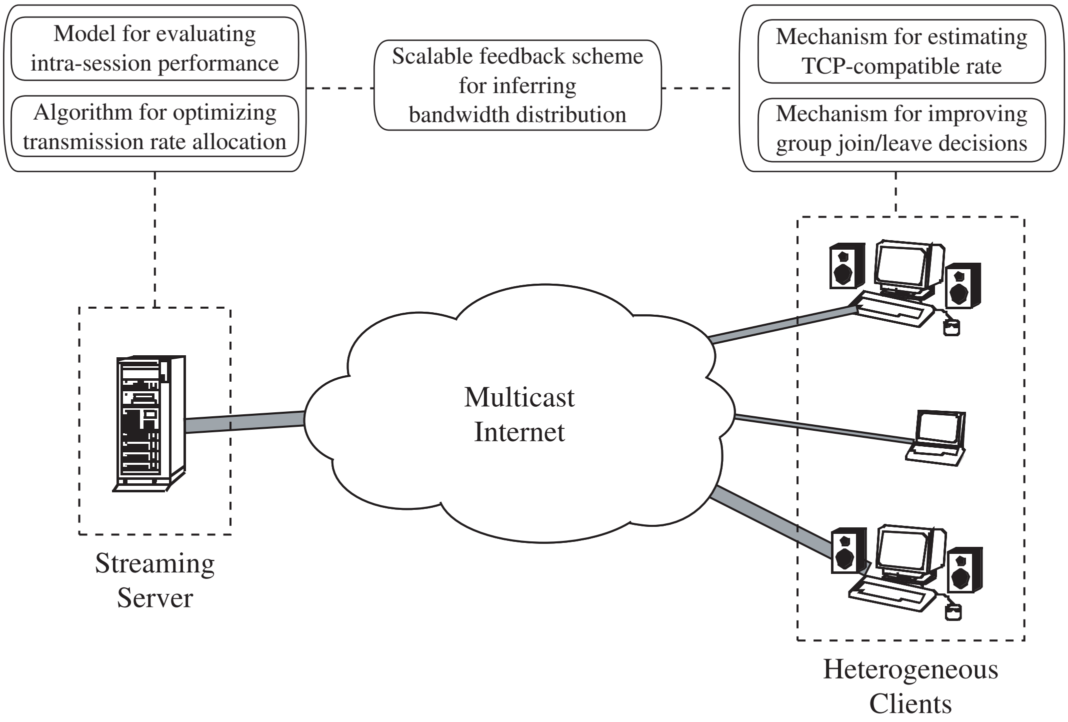

Model for evaluating intra-session performance
Algorithm for optimizing transmission rate allocation
Scalable feedback scheme for inferring bandwidth distribution
Mechanism for estimating TCP-compatible rate
Mechanism for improving group join/leave decisions
Multicast Internet
Streaming Server
Heterogeneous Clients

Figure 1.1: Illustration of scenario and contributions.

1.5 Structure of the Thesis

The remainder of the thesis is organized as follows. Chapter 2 introduces the context of the thesis by giving an architectural overview and introducing the technical background of the underlying network model and media characteristics. The chapter completes with a survey of related work in the area of multicast rate adaptation, stream optimization, and feedback control.

In Chapter 3 we first discuss a model for capturing the satisfaction of a receiver participating in a multicast streaming session, and a global session performance metric is derived. These build the basis for the following investigation of an algorithm for optimal allocation of transmission rates. Finally, we conduct simulations and evaluate the performance of multi-rate and adaptive rate allocation schemes.

Chapter 4 focuses on the development of a scalable feedback scheme for collecting information about the bandwidth distribution of the receiver population. We study a light-weight approach based on sampling and a model-based approach that builds on measurement results available from the Internet community. Using simulations we analyze the performance and compare both of the approaches for several theoretical and empirical receiver distributions.

In Chapter 5 we develop and study an algorithm for inferring the fair share, that is, the TCP-compatible rate of a multicast receiver. By means of network simulations we show that naively adopting state-of-the-art mechanisms from the unicast to the multicast case does not always provide the expected results. After the analysis of the underlying model we devise a modified estimator and evaluate its behavior.

Based on the developed fair share estimator, in Chapter 6 we investigate a mechanism for improving multicast group join and leave decisions, which avoids oscillations while being responsive to congestion indications. We implemented our algorithms and mechanisms in a protocol framework using the *ns-2* network simulator environment. Through simulations we evaluate the performance of our solution and compare it to one of the most prominent layered multicast congestion control schemes.

We conclude our work and present directions of future work in Chapter 7.

Figure 1.2 provides a roadmap for the thesis following its structure and contributions.

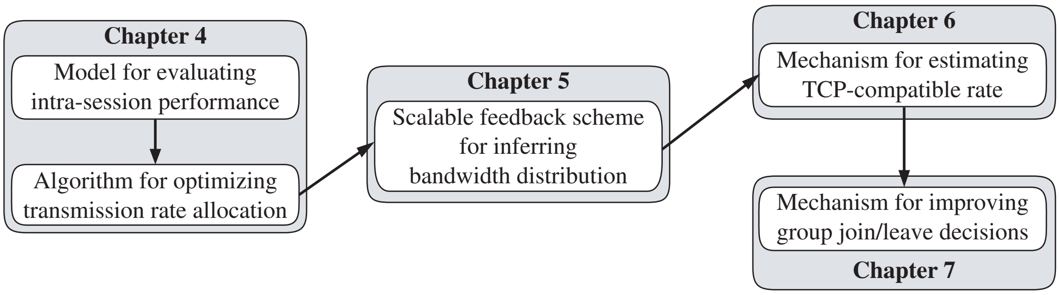

Chapter 4
Model for evaluating intra-session performance
Algorithm for optimizing transmission rate allocation
Chapter 5
Scalable feedback scheme for inferring bandwidth distribution
Chapter 6
Mechanism for estimating TCP-compatible rate
Mechanism for improving group join/leave decisions
Chapter 7

Figure 1.2: Roadmap of the thesis.

Chapter 2

Background

The major focus of this chapter is on providing background information and setting the context of our work. Following the introduction of the terminology, we describe our end-to-end model of a multi-rate multicast adaptation scheme. We emphasize that the goal of this thesis is the investigation and development of mechanisms for rate adaptation that can be used as modular building blocks. The design and specification of a protocol architecture or framework, however, is out of scope.

After the overview of the general components of a multi-rate multicast adaptation scheme, we introduce the underlying network model assumed for all of our work, which is based on IP multicast. The subsequent section focuses on media characteristics. It provides an overview of the principles of scalable video coding techniques and introduces the concepts of video utility functions and video quality metrics. Finally, we complete the chapter with a survey of existing work directly related to the thesis.

2.1 Overview

The Internet is an interconnected "network of networks" based on the IP architecture. Globally, it provides only a best-effort service so that data flows actively compete for network resources. A fundamental requirement crucial for the stability of the Internet is cooperativeness amongst competing flows. Consequently, multicast streaming solutions have to incorporate mechanisms to prevent unfairness to other flows. They are expected to adopt the "social" rules and behave cooperatively by appropriately reacting to congestion signals.

The natural end-to-end approach for preventing a streaming application from causing and contributing to congestion is by adaptation of its transmission rate(s). Potential multicast rate adaptation schemes are traditionally classified into two categories: sender-based (single-rate) schemes and receiver-based (multi-rate) schemes. In the sender-based schemes there is usually a single multicast channel for dissemination of the video so that the latter is transmitted to all receivers at the same rate. It is the responsibility of the sender to perform the rate control functionality by adapting the transmission rate, for example, to meet the conditions of the most congested data path (e.g., TFMCC [WH01]). The receiver-driven approach relies on multi-rate multicast transmission. Thereby, video data is provided to the members of a session in several quality levels each implying a certain data rate. In a layered transmission scheme the resulting data rate is determined by the number of multicast channels that a host is subscribed to. The underlying distribution tree of a multicast channel branches to a certain path only if an active receivers is subscribed on this path. Thus, each receiver is in charge of performing rate adaptation by adjusting its subscription level, that is, the number of received layers (e.g., RLC [VRC98]).

Only recently, a few hybrid schemes have evolved that combine the aforementioned paradigms. These schemes employ receiver-based mechanisms to adjust the rate of each receiver to the latter's network conditions in a short term. The source is responsible for adaptation of the provided data rates on a longer term to better match the global conditions of the active receiver set. While this hybrid approach generally provides a promising alternative to the traditional approaches, current solutions are still at their infancy and can be significantly improved by more efficient and stable sender-side as well as receiver-side algorithms and mechanisms. Contributing to the design of these algorithms and mechanisms is the goal of our work.

2.2 Terminology

Here we introduce some important terms that we use throughout our work:

Multicast group. The network-centric term refers to the IP concept of a multicast group. The network service provided by IP multicast is a *channel* identified by its IP multicast group address that is mapped onto a multicast distribution tree. In the context of layered transmission, each video layer is sent to an individual multicast group.

Session. A multicast session includes a collection of end hosts that communicate using a particular set of IP multicast group addresses. This definition follows the Light-Weight Session (LWS) model [Jac94] that underlies the MBone tools. Therein, multiple data flows using different multicast addresses to transport a portion (layer) of the same video belong to the same session. The term *intra-session* is then used to refer to the relation within the same session, while the term *inter-session* refers to the relation between different sessions.

Transmission group. The term transmission group refers to a set of receivers of a multicast session that are subscribed to the same set of multicast groups. Thus, the streaming media is being transmitted to all members of the transmission group at the same quality level. In a layered multicast session, transmission groups are distinct.

Subscription level. A subscription level corresponds to the subset of multicast groups that the receivers of a transmission group are subscribed to. In the context of layered transmission, the subscription level of a receiver determines the number of the highest layer and the quality level being transmitted to the receiver. The subscription level is changed by joining and leaving multicast groups.

Rate allocation. We use the term rate allocation to refer to the source process of partitioning the streaming data for the transmission over multiple multicast channels. The scalable-encoded video stream is partitioned into several sub-streams each mapped onto a single channel. Thereby, the transmission rate of each sub-stream is determined by the data rate allocated to each channel by the server.

2.3 A Hybrid Rate Adaptation Model

In this section, we give an overview of the functionalities and the interactions of the mechanisms developed in the course of the thesis. As already mentioned in the preceding introductory section, the mechanisms are particularly designed to provide functionalities for empowering hybrid rate adaptation schemes. Figure 2.1 depicts our model, whereby the gray boxes denote the target components our work provides solutions for:

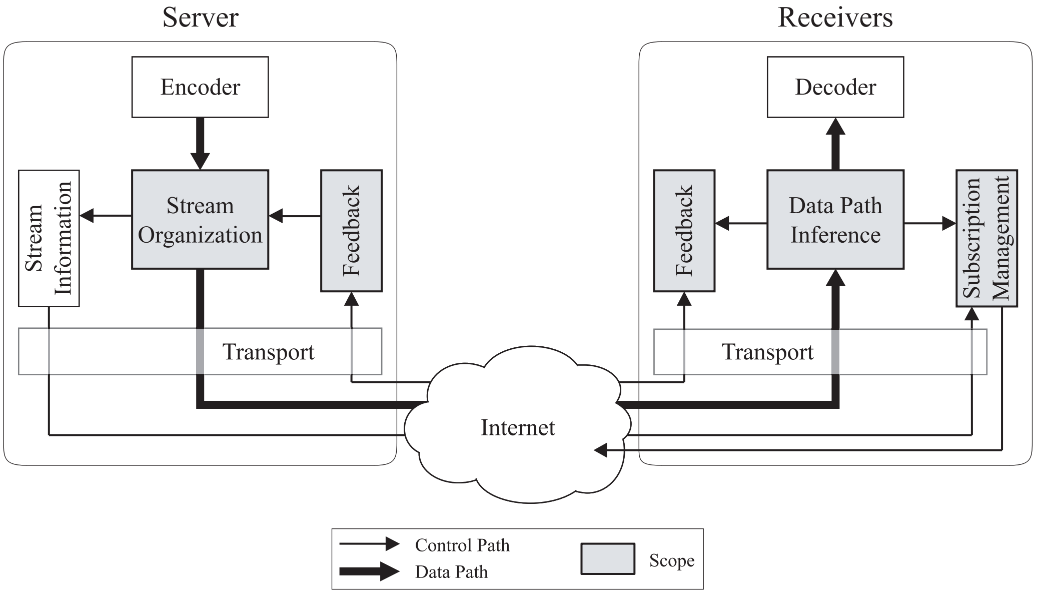


Figure 2.1: Our model of a hybrid multi-rate multicast rate adaptation scheme.

1. **Stream organization.** The server divides the scalable-encoded video into a number of cumulative layers. Each additional layer received by a host improves the latter's received media quality. In our approach, data rates allocated to the layers are dynamically adjusted according to receiver feedback in order to optimize the overall satisfaction of the users. The underlying optimization metric and the optimization algorithm are discussed in Chapter 3.

2. **Feedback control.** Stream organization is performed based on the distribution of receiver bandwidth capabilities. In order to gather information about the global network conditions, the server collects status reports from the receivers. To limit the feedback traffic and increase the scalability of our scheme, we employ a feedback mechanism based on probabilistic sampling that is discussed in Chapter 4.

3. **Data path inference.** The individual data path conditions are inferred by the corresponding receiver. Therefore, the latter detects packet losses and estimates the round-trip time exploiting the sequence numbering and time stamping functionalities of the underlying transport protocol. The measured loss rate and round-trip time are used to continually calculate an estimate of the actual TCP-fair rate. The estimation model and technique are crucial

components, which are the subject of Chapter 5.

4. **Subscription management.** Based on the estimated TCP-fair share and the actual data rates provided by the server, each receiver chooses a subscription level that matches best its actual network conditions. In order to avoid frequent fluctuations that would disturb the user perception but still provide for the necessary degree of responsiveness to congestion indication, a suitable subscription management strategy is presented in Chapter 6.

The white boxes in Figure 2.1 represent components that are out of the research scope of this thesis. The distribution of the session information, that is, the data rates allocated to each layer, is straightforward. This information can be encapsulated either in feedback polling messages, in dedicated announcement messages, or piggybacked on data packets. The network model providing data transport mechanisms is based on the IP multicast architecture and will be discussed in Section 2.4. Finally, video is assumed to be encoded with recent coding schemes that allow for scaling the quality and data rate of the media. Since the relation between data rate and media quality is exploited in our optimization metric, we summarize the characteristics of scalable-encoded video in Section 2.5.

2.4 The Network Model

The work in this thesis builds on the IP multicast service model that has been originally proposed by Steve Deering in 1989 [Dee89]. We present a short overview of this network model in order to provide the necessary context for our work. For a comprehensive overview and a detailed discussion of the IP multicast technology, we refer to many excellent books and articles (e.g., [Kos98], [WZ99], [HC99], [DLL+00], and [BCHC02]).

2.4.1 The Internet Protocol

The Internet Protocol (IP) architecture defines mechanisms and protocols for use in interconnected packet-switched networks. IP provides means for transmitting blocks (datagrams) mapped onto packets between communicating end systems or hosts. Each of the latter has one or more network interfaces that attach it to one or more networks. Thereby, each interface is assigned a globally unique identifier, that is, an IP address.

The two basic functions implemented in IP are addressing and fragmentation. When a packet arrives on an incoming interface of an IP router, the latter examines the destination address carried in the IP header. Since each router maintains a routing table that maps destination addresses to outgoing interfaces, it can locate the proper route and forward the packet. If a packet is larger than the Maximum Transmission Unit (MTU) of the underlying network, it is fragmented into smaller packets and reassembled at the destination host.

The Internet provides only a best-effort service and does neither guarantee that packets will reach their destination within a particular period of time nor that they will reach their destination at all. Packets might be delayed in router queues as the system load increases. In the worst case of *congestion*, packets can be dropped since queue buffers might overflow. Hence, if applications require reliable delivery of data, the transport or application layer has to provide the functionality. Furthermore, the applications are expected to adopt the "social" rules of the Internet and behave cooperatively by reacting to congestion signals and adapting their transmission rates properly.

2.4.2 IP Multicast

The traditional IP architecture provided only *unicast* transmission, where packets are delivered from the source to a single destination in a point-to-point communication context. For multipoint communication, a source has to send an individual copy of the packet to each recipient. Obviously, this is extremely inefficient in terms of source and network resource usage. A much more efficient technique has been proposed by Steve Deering [Dee89] in the IP multicast service model. It is based on replication of packets only at fan-out points in the network so that at most one copy of each packet is transmitted over a link. Figure 2.2 illustrates the principle difference of multipoint communication based on IP unicast and on IP multicast, respectively.

Instead of forwarding packets along multiple paths, in IP multicast they are forwarded to all destinations along a single *distribution tree* rooted at the data source or a "rendezvous point" (core). To compute spanning trees from the latter to all receivers, a number of routing algorithms and protocols[1] exist, for example, Distance Vector Multicast Routing Protocol (DVMRP) [WPD88], Multicast Open Shortest Path First (MOSPF) [Moy94], Core Based Tree (CBT) [Bal97], and Pro-

---

[1] A thorough discussion on multicast routing protocols is provided in [PR02].

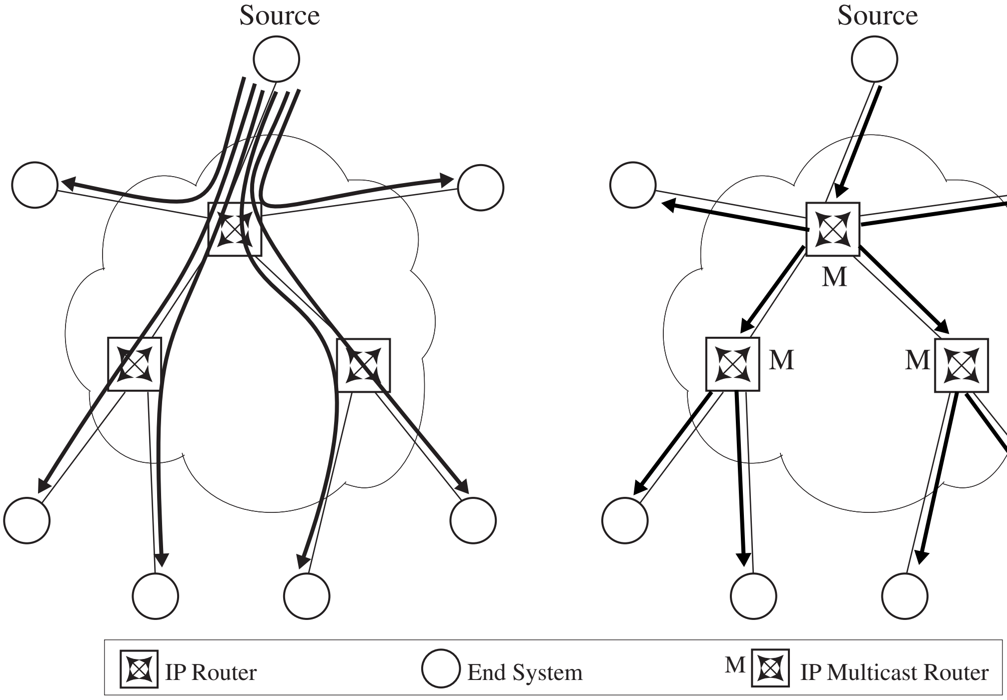


Figure 2.2: Data dissemination over IP unicast (left) and IP multicast (right) multipoint communication.

tocol Independent Multicast (PIM) [EFH$^+$98]. Because the source address (or the address of the core) identifies the spanning tree, it can be used for routing table indexing. The routing decision is based on the source address, for which the entry in the routing table provides a set of outgoing interfaces. As a result, routing decisions in multicast are determined by the source address in contrast to unicast where routing is determined only by the destination address.

The level of indirection provided by the host group abstraction is a key feature of IP multicast. More specifically, while unicast packets are routed based on the destination address included in the packet header, a multicast source generally does not need to have explicit knowledge about receivers and vice versa. Instead, the source pushes packets to one or more IP group addresses and receiver-driven group membership is utilized for implicit control of data distribution. That is, interested receivers have to signal their interest in joining and leaving multicast groups to the network by means of Internet Group Management Protocol (IGMP) messages.

Source-Specific Multicast

Although it has been available through the experimental IP Multicast Backbone (MBone) for a number of years, IP multicast has seen slow commercial deployment in the Internet. One of the main reasons that stalled its widespread use is attributed to the original Any-Source Multicast (ASM) service model that has been developed for supporting a vast class of applications. The resulting complexity makes the architecture unstable, which has been noticed by major carriers [DLL^{+}00].

Remedy is provided by the Source-Specific Multicast (SSM) service model, which recently has been adopted by the IETF. SSM is a much simpler service model that attempts to solve many of the deployment problems of ASM including protocol complexity, inter-domain scalability, and security weaknesses [ABD01]. It is particularly suited for the current set of one-to-many applications that drive multicast deployment, such as video streaming as being considered in the context of this work. Hence, the thesis builds on the IP SSM service model, which is expected to further accelerate the commercial deployment of multicast to become a mature network service [DLL^{+}00].

2.4.3 Data and Message Transport

While IP multicast provides point-to-multipoint channels on the network layer, transport protocols for carrying the data and messages between the members of a multicast session have to be placed on top of it. The development of new transport protocols is not the goal of the thesis. We emphasize that our intention is to develop mechanisms that can be integrated as building blocks into different protocols frameworks. Thus, since the focus of the thesis is on the dissemination of real-time video, Real-Time Transmission Protocol (RTP) is considered as the example framework for encapsulating the media streams on top of User Datagram Protocol (UDP).

RTP defines much of the protocol architecture necessary for video transmission over multicast packet networks. For the estimation of the fair share discussed in Chapter 5, it provides the necessary time stamping and sequence numbering mechanisms. Furthermore, it is accompanied by a flexible control protocol, the Real-Time Transmission Control Protocol (RTCP). The latter can be utilized for application-specific control messages such as the feedback request and response messages (status reports) in Chapter 4. Furthermore, the sender can inform the receivers of the session about the actual data rates of the multicast groups using

RTCP messages. This is necessary when using the sender-side rate adaptation algorithm developed in Chapter 3 in order for receivers to make appropriate subscription decisions in Chapter 6. Figure 2.3 summarizes the discussed protocol stack underlying our work.

<table>
<tr><td>Video
Data</td><td>Layering
Info</td><td>Feedback
Requests</td><td>Status
Reports</td></tr>
<tr><td>RTP</td><td colspan="3">RTCP</td></tr>
<tr><td colspan="4">UDP</td></tr>
<tr><td colspan="3">IP Multicast</td><td>IP</td></tr>
</table>

Figure 2.3: Protocol stack of the thesis.

2.5 Media Characteristics

The hybrid rate adaptation model underlying this thesis is based on multi-rate transmission, specifically layered transmission. Therefore, features of scalable-encoded video in general and FGS - encoded video in particular are exploited. FGS is a modern variant of scalable video enabling sender-side adaptation functionality. The usage of scalable coding schemes allows for efficient streaming of a single video stream at different bit rates and quality levels.

In the following, we first give a short overview of scalable video and its mapping to a layered transmission scheme. Subsequently, the modeling of user satisfaction by means of utility functions is introduced followed by a discussion of video quality metrics. These concepts are employed for concrete instantiations of the utility fairness metrics developed in Chapter 3.

2.5.1 Scalable Video

Traditional video coding schemes, such as MPEG-1 or H.261, exploit intra-frame as well as inter-frame redundancy and irrelevance to reduce the data volume of the encoded video. The objective of these schemes is to optimize video quality at a given bit rate. For dissemination over dedicated infrastructures, such as cable and satellite TV broadcast systems, the stream rate ideally matches the channel capacity. However, pre-encoding media streams at an optimal rate is generally not possible for real-time transmission over packet-switched networks. Since resource availability in open platforms such as the Internet dynamically changes and is not known a priori, the inelastic demand of streaming applications poses a severe problem.

In order to overcome the aforementioned limitations of traditional coding schemes, modern coding schemes providing scalability have been adopted by the standardization bodies. These schemes produce bit streams decodable at different bit rates. Scalability can thereby be categorized into four classes [PE02]: spatial scalability, temporal scalability, Signal-to-Noise Ratio (SNR) scalability, and object-based scalability. Spatial scalability offers the functionality to decode the video at different spatial resolutions, while temporal scalability allows the adjustment of the frame rate. SNR scalability allows to decode the signal at the same spatial and temporal resolution at different quality levels. This is achieved, for example, by layered quantization of the component values of the Discrete Cosine Transform (DCT) [AMV96]. The fourth class of scalability allows for decoding of a subset of the audio-visual objects composing a video scene.

The most prominent approach making use of scalability is cumulative layered coding. With this approach the video signal is split into one base layer and one or more enhancement layers. The base layer contains the base information that is necessary to decode the video at a minimum quality. Each enhancement layer contains additional information that increase the quality of the reconstructed video signal. In the cumulative approach, the enhancement layers are hierarchically organized. To reconstruct the information included in layer i, all layers $(1, \ldots, i-1)$ have to be available.

The concept of scalable coding was fist introduced in MPEG-2 and H.263, which basically allow for a base layer and a single enhancement layer. Further extension in H.263+ and MPEG-4 provide support for a higher number of layers with predetermined bit rates. In a layered multicast transmission scheme each video layer is then mapped onto a dedicated multicast channel allowing users to control their rate and quality by the choice of the subscription level. Figure 2.4 illustrates the principle of layered video and its mapping onto a layered transport context.

Recently, MPEG-4 further extended video scalability by adopting FGS [Li01]. With this coding technique, the video signal is encoded into a fixed-rate base layer and an enhancement layer that can be truncated at an arbitrary bit rate. Video encoded with the FGS scheme provides flexibility and enables multicast rate adaptation solutions to stripe and dynamically (re-)partition the enhancement into multiple transport layers. The principles of FGS and its layered transport is depicted

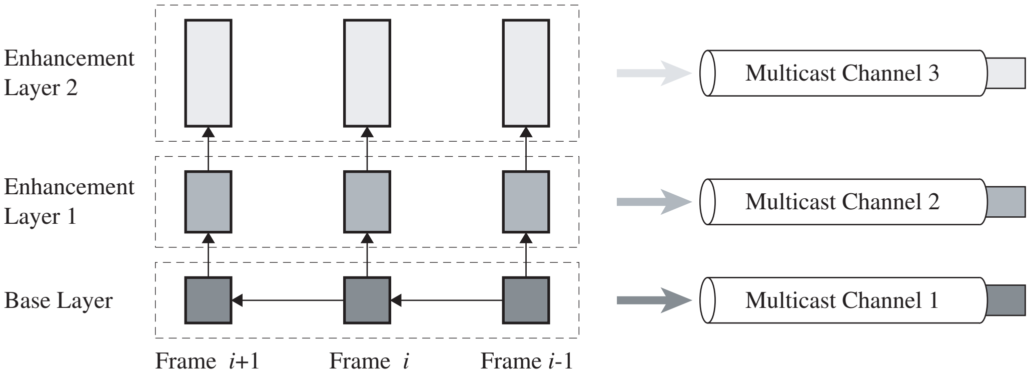


Figure 2.4: Layered video and its mapping onto multicast transport channels.

in Figure 2.5. In Chapter 3 of this thesis, we exploit these powerful scaling features of FGS for optimization purposes.

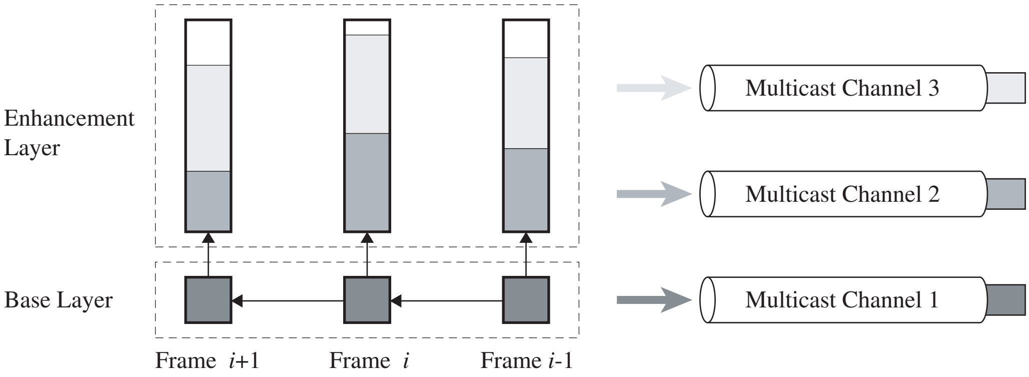


Figure 2.5: FGS-encoded video and its mapping onto multicast transport channels.

2.5.2 Video Utility Functions

The concept of utility functions has been used in network research as a theoretical abstraction of application demands for network pricing and optimization of resource allocation [She95]. This approach originates from the demand that network performance should be evaluated solely in terms of the degree to which it satisfies the service requirements of user applications. Hence, utility functions also provide means for modeling the user satisfaction of streaming video over packet-switched network.

In the modeling process, users are assumed to have a utility function that maps from a given network QoS q to a level of satisfaction $u(q)$. Generally, the QoS depends on several parameters such as the data rate, delay, jitter, and loss. In the context of this thesis, however, we consider the data rate as the primary QoS factor. In contrast to interactive applications (e.g., IP telephony), the majority of video streaming applications have soft real-time requirements[2] [SN04]. Furthermore, buffer space at the receiving clients is usually dimensioned large enough to deal with jitter. And finally, for our discussion we assume a perceivable loss level to occur merely when the communication system gets overloaded, that is, the demand on bit rate exceeds the system's bandwidth[3] capacity. In the resulting model the user utility u reduces to a function of the bit rate r.

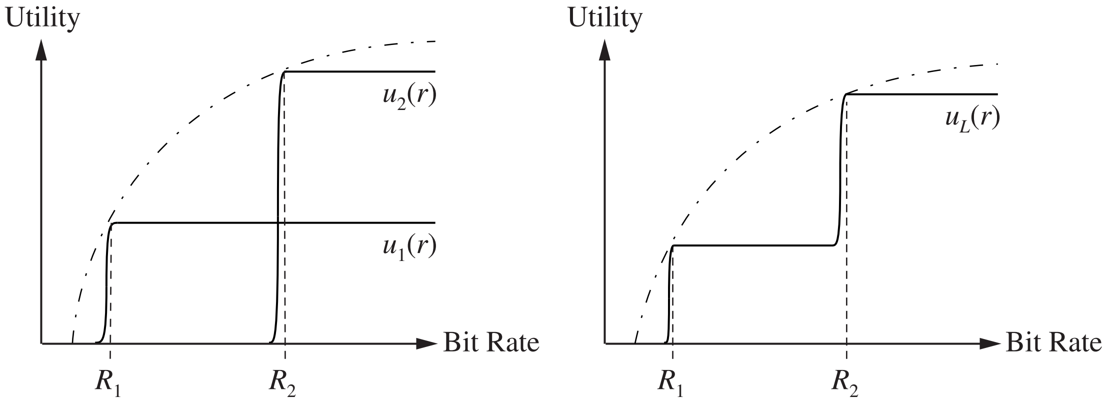
For two media streams encoded with traditional schemes at different bit rates, Figure 2.6(a) depicts their utility functions u_1 and u_2 as a function of the received bit rate[4]. The received bit rate is determined by the available bandwidth of a receiver's data path and if it is below the data rate of the video, the transmitted stream will experience a significant number of packet losses. As a consequence, the decoded video will suffer severe distortions providing the user with hardly any utility. Once the transmission rate matches the target rate of the encoded video, the latter is decoded and displayed at a quality level that is by the encoding parameters. (We discuss the relation between video quality and data rate in Section 2.5.3). For transmission rates beyond the target bit rate of the video, the marginal gain in user utility reduces to zero. A scalable-encoded video, however, can be transmitted and decoded at different rates so that the utility can be gradually increased, as illustrated for a two-layer video in Figure 2.6. In the above examples, utility is assumed to be determined by the rate-distortion curves of the coding scheme. We discuss rate-distortion functions in the context of video quality in the following section.

---

[2] Note that we consider streaming of educational, informational, and entertainment content over multi-service networks. Time critical applications, such as tele-surgery, are not considered—their stringent timing requirements cannot be met by today's Internet.

[3] Throughout this document we use the term bandwidth and bit rate synonymously to refer to the data rate in bits per time unit. This is common practice in Internet research community. Thus, the usage of the term bandwidth differs from the physical bandwidth, which denotes a frequency span expressed in Hertz (Hz).

[4] Note that without the loss generality, we neglect packetization overhead for the simplification of the discussion.



(a) fixed-rate encoded video (b) scalable-encoded video

Figure 2.6: Illustration of video utility curves.

2.5.3 Video Quality Metrics

Appropriate models for capturing the user utility[5] as a function of the video bit rate are the subject of this section. In particular, we consider that the bit rate and the level of user satisfaction in terms of video quality do not exhibit a linear relationship [KTPE99, Win99, Zin03]. Thus, this issue is addressed in the following discussion on the general dependency of the quality of streaming video on the decoding bit rate, considering specifically the characteristics of FGS-encoded video.

It is out of the scope of this thesis to contribute to the area of video assessment and quality metrics. Our interest is rather in a simple mathematical approach to approximate the perceived quality based on existing work. This is necessary for the quantitative evaluation performed in Chapter 3.

Quality Assessment

For the measurement of video quality there are two approaches: subjective methods and objective methods. Subjective quality assessment methods measure the overall perceived quality and are conducted by human subjects. That is, a group of users is supposed to view processed video sequences in order to rate their quality. Recommendations for subjective video quality assessment of the International Telecommunications Union (ITU) include specifications on how to perform differ-

---

[5] We use the terms user utility and receiver utility interchangeably as a metric for the user satisfaction.

ent types of subjective tests, and the grading scale [ITU02, PW03]. But the test methods present sever limitations: (a) they require stringent environments and cannot be automated; (b) they are very time-consuming and consequently expensive; (c) they cannot feasibly be used in real time. Obviously, automated evaluation methods mimicking the human visual system would provide a very powerful tool.

An accurate model for the human visual system must consider all relevant factors including spatial and temporal resolution, brightness, contrast sharpness, colorfulness, viewing distance, viewing size, human spatial-temporal contrast sensitivity, fluctuations, and other factors. Developing such a model is obviously a complex undertaking, which has been subject to intensive research for many years. A first comprehensive metric based on a spatio-temporal model of the human visual system for the above purpose has been proposed as early as 1982 in [LB82]. Other models and metrics followed, for example in [Gir88, Wat90, TGP98, Moh03], and in the past few years there has been increasing interest in perceptual video quality assessment. However, due to the complexity of the human visual system yet there exists no objective measurements that correlate well with the human perception and that are feasible to perform in real time [Win99, VQE00, Str02, WOZ02].

Driven by the above observations, within this thesis we follow the common practice and resort to objective methods for accessing the quality of a video signal based on pure mathematical measurements.

Rate-Distortion Relations and Metrics

For objective assessment of video quality, distortion measures and distortion-based quality measures are defined and used. Distortion metrics evaluate the difference of the corresponding original and reconstructed signal, calculating values that increase with the signal difference.

Let $x[k]$ denote the original signal of a video frame consisting of K samples, and $y[k]$ the reconstructed signal. The distortion is then generally formulated as a function of these both sequences:

$$D = D(x[k], y[k]). \tag{2.1}$$

For a better distinction of the sequence $x[k] = (x_1, \ldots, x_K)$ and its elements, in the following we denote the individual symbols with x_i, respectively y_i for the sequence $y[k]$ and its elements.

The relation between the signal quality and the bit rate of a video are expressed through rate-distortion (R-D) curves, which heavily depend on the underlying content and the encoding scheme, and may even vary significantly from frame to frame of a single video sequence. Most commonly the Mean Square Error (MSE) and the Peak Signal-to-Noise Ratio (PSNR) are used as metrics to access the distortion respectively the signal quality, both calculated for the luminance signal only. The MSE is defined as:

$$\mathrm{MSE} = \frac{1}{K} \cdot \sum_{i=1}^{K} (x_i - y_i)^2 \,, \tag{2.2}$$

and the PSNR maps it to a signal-related quality measure

$$\mathrm{PSNR} = 10 \cdot \log_{10} \left(\frac{x_{pp}^2}{\mathrm{MSE}} \right) \quad \Leftrightarrow \quad \mathrm{MSE} = \frac{x_{pp}^2}{10^{\sqrt[10]{\mathrm{PSNR}}}}, \tag{2.3}$$

where x_{pp} denotes the peak-to-peak value, which equals 255 for the usual 8-bit representation of the video luminance signal component.

Rationale for Using Standard Measures

In most cases there exists a relation between the quality experienced by the viewer and the PSNR values of the corresponding video, although the latter ratio is widely criticized for not correlating well with perceived quality measured by subjective assessment. In [VQE00] the Video Quality Experts Group (VQEG) concluded that the performances of proposed objective quality models are statistically indistinguishable from that of PSNR. Consequently, it is common practice to compare and evaluate coding schemes in the PSNR domain.

A popular statistical model for DCT-based video is that of a Gaussian source with mean $\mu_x = 0$ and variance σ_x^2. This model leads to tractable results in information theory [CT91] for the relation between the video bit rate R specified in bits per pixel (bpp) and the distortion D:

$$R(D)_{Gauss} = \frac{1}{2} \log_2 \left(\frac{\sigma_x^2}{D} \right) \quad \Leftrightarrow \quad D_{Gauss} = \frac{\sigma_x^2}{2^{2R}}. \tag{2.4}$$

Substituting in Equation 2.3 the distortion with the one obtained from the statistical model in Equation 2.4 leads to a linear function for the signal quality:

$$\mathrm{PSNR}(R)_{\mathrm{Gauss}} = 10 \cdot \log_{10}\left(2^{2R} \cdot \frac{x_{pp}^2}{\sigma_x^2}\right) \quad \Rightarrow \quad \frac{d\mathrm{PSNR}(R)_{\mathrm{Gauss}}}{dR} = \mathrm{const.} \qquad (2.5)$$

According to Equation 2.5 the increase in quality is constant for a fixed increase of the rate. Recall that the equation is only valid for the assumption of a Gaussian source and provides upper bounds on achievable quality. Real R-D curves, as those for FGS-encoded video presented and analyzed in [dCRR02] and [dCRR03], exhibit a rather non-linear relationship between the bit rate and the PSNR. This observation holds in particular for lower bit rates and led to the development of quadratic and square-root models [CZ97, DL03].

Studies based on subjective assessment [Zin03] and objective perception metrics [KTPE99] indicate a similar but more pronounced level of concavity of video quality curves than usually observed in PSNR traces. That is, the marginal utility of a video is decreasing faster with an increasing bit rate. Since a well-founded function capturing the above correlation precisely has not been manifested yet, a simple approximation can be obtained by formulating the user utility as a linearly decreasing function of the MSE index, which can be straight-forwardly obtained from existing PSNR values[6] (see Equation 2.3).

Since there is still no consensus on a simple and appropriate metric for accurately modeling user utility, for our work we derive utility functions based on the above introduced mathematical measurements, namely the PSNR and the MSE, respectively.

FGS-fitted Rate-Distortion Model

Extensive studies of FGS-encoded video and a large library of corresponding traces in the PSNR-domain recently became available [dCRR02, dCRR03]. However, for the purpose of fine-grained adaptation and optimization of multicast transmission rates by means of utility function based on signal distortion and quality, a closed-form R-D model for the latter is required.

Recently an accurate mathematical distortion model for the enhancement layer has been presented in [DL03]. It provides a powerful yet simple tool modeling the distortion of the FGS-residue $D(r_{enh})$ as a function of the enhancement layer

---

[6] It is common practice in the area of quality adaptation for streaming video to resort to the distortion measure for optimization purposes [WFLG00, CH01, CS02, DLR03].

rate r_{enh}. This is sufficient to describe the PSNR and MSE of a video for a fixed base layer with rate r_{base}, since the overall distortion depends exclusively on the distortion introduced by truncating the enhancement information. We explain this by means of Figure 2.7.

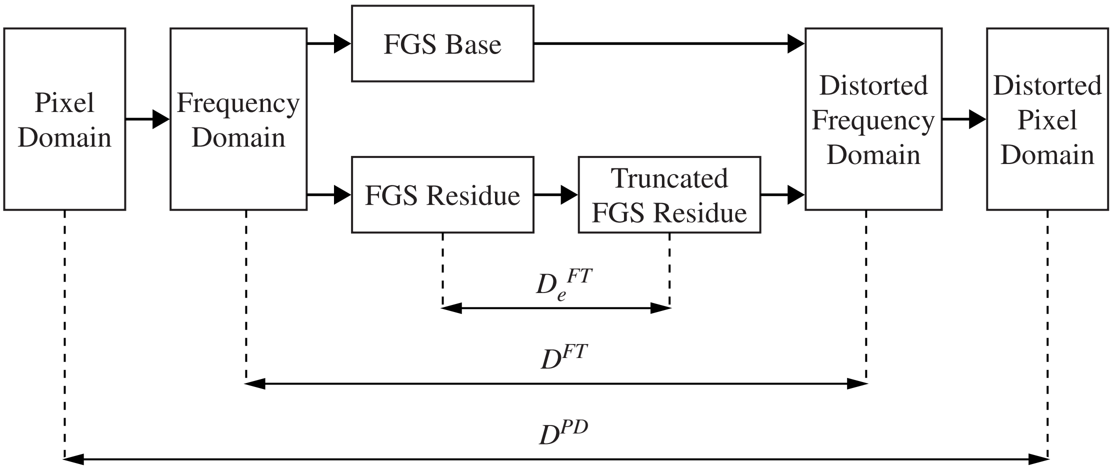


Figure 2.7: Distortion model for FGS-encoded video.

The original video frames are transformed by means of the DCT from the pixel domain (PD) to the frequency domain (FD), and vice versa by means of the inverse transform. Although the DCT is an orthogonal transform, real encoder and decoder introduce quantization round-off errors, which however in practice are marginal and considered negligible:

$$D = D_{PD} \approx D_{FD} \quad \Rightarrow \quad x_i = x_{PD,i} \approx x_{FD,i}, \quad y_i = y_{PD,i} \approx y_{FD,i}. \tag{2.6}$$

The subscript PD and FD denote the concerned quantity in the picture domain and frequency domain, respectively. Let us refer to the base and enhancement elements of the original signal with $x_{base,i}$ and $x_{enh,i}$, respectively with $y_{base,i}$ and $y_{enh,i}$ to the corresponding elements of the reconstructed signal. Since for rate adaptation purposes only the enhancement layer is truncated, the elements of the reconstructed base layer are not distorted, that is, $y_{base,i} = x_{base,i}$. For the distortion it then holds that:

$$D = \sum_i (x_i - y_i) = \sum_i (x_i - (x_{base,i} + y_{enh,i})) = \sum_i (x_{enh,i} - y_{enh,i}) = D_{enh}. \quad (2.7)$$

Obviously, the distortion of a FGS-encoded signal is a function of the enhancement layer only. Thus, from the square-root model closed-form representations for the PSNR and MSE can be defined as follows

$$\begin{aligned} \mathrm{PSNR}(r) &= \nu_1 \cdot (r - r_{base}) + \nu_2 \cdot \sqrt{r - r_{base}} + \nu_3 && (2.8) \\ \mathrm{MSE}(r) &= \frac{x_{pp}^2}{10^{\frac{\nu_1 \cdot (r - r_{base}) + \nu_2 \cdot \sqrt{r - r_{base}} + \nu_3}{10}}}, && (2.9) \end{aligned}$$

where r and r_{base} denote the cumulative rate respectively the rate of the fixed base layer such that $r = r_{base} + r_{enh}$ and $r \geq r_{base}$. Furthermore ν_1, ν_2 and ν_3 are video sequence specific parameters, where ν_3 is determined by the base layer quality $\mathrm{PSNR}(r_{base}) = \nu_3$.

2.6 Related Work

This section surveys existing work related to the thesis. We discuss end-to-end multicast control schemes that rely on the collaboration of the sender and the receivers or on collaboration among receivers only. They are generally referred to as congestion control schemes, whereby control by rate adjustment represents the natural and dominant form for streaming applications. Subsequently, work related to sender-based optimization of the transmission rates is surveyed followed by feedback control schemes.

2.6.1 Rate Adaptation

TCP and its congestion avoidance and control mechanisms have enabled a rapid growth of diverse unicast applications. While the usage of corresponding mechanisms for multicast is also required to provide for cooperativeness and social behavior, their design has proved to be a far more difficult and complex problem. It has been recognized that multicast applications have a much wider range of requirements, which a single, generic protocol cannot meet [Obr98]. Hence, many researchers have proposed various solutions for multicast transport and control since the introduction of IP multicast [Dee89]. In the following, we focus the discussion on more recent approaches that are partly covered in the survey of TCP-friendly congestion control of Widmer et al. [WDM01]. Excellent surveys of earlier solutions have been published by Diot et al. [DDC97] and Obraczka [Obr98].

Single-Rate Multicast

One of the earliest schemes for multicast rate control has been developed by Bolot et al. [BTW94] in the context of the INRIA Videoconferencing System (IVS) [Tur94]. It belongs to the sender-driven, single-rate schemes and adapts its transmission rate to meet the network conditions of the worst receiver. Therefore, Bolot et al. developed a scalable feedback suppression mechanism for estimating the group size and detecting the worst network state. Instead of requiring feedback of all receivers, DeLucia and Obraczka [DO97] proposed to use representatives of receiver groups. Using ACK equivalent and NACK equivalent feedback messages, the representatives inform the source about congestion state. The source then uses the feedback to control its transmission rate with a Multiplicative Increase Multiplicative Decrease (MIMD) algorithm. The concept of representatives has also been proposed by Rizzo in Pragmatic General Multicast Congestion Control (PGMCC) [Riz00], where a single so-called acker dictates the source for rate adaptation by means of ACKs. The underlying window-based control strategy is similar to TCP's Additive Increase Multiplicative Decrease (AIMD), thus, causing frequent rate fluctuations and significantly influencing the perceived quality of streaming media.

A scheme that overcomes the shortcoming of window-based adaptation is TCP-Friendly Multicast Congestion Control (TFMCC) proposed by Widmer and Handley [WH01]. Similar to PGMCC the transmission rate is controlled by the feedback of the limiting receiver. While the control parameter in PGMCC is the sender's congestion window, the rate-based control of TFMCC targets at smooth transmission of streaming media. The proposed scheme is basically a multicast extension of TCP-Friendly Rate Control (TFRC), the state-of-the-art equation-based congestion control scheme for unicast transmission. For the estimation of its TCP-fair rate using a mathematical model, a receiver has to measure its round-trip time and loss rate. With the calculated value of the fair share it instantiates an exponentially weighted feedback timer. For feedback suppression purposes, feedback timers are biased in favor of receivers experiencing a higher congestion state, and timers are canceled by reception of other receiver's feedback message. While providing a very promising approach for multicast streaming, TFMCC suffers the common limitations of single-rate schemes. With the provision of only a single rate to all receivers it cannot accommodate the latter's heterogeneity in terms of bandwidth capability.

Multi-Rate Multicast

Multi-rate multicast transmission is an elegant way to address the heterogeneity challenge. One of the first working examples was Receiver-driven Layered Multicast (RLM) developed by McCanne et al. [MJV96]. RLM utilizes a layered transmission scheme to deliver streaming video to multicast receivers of the same session at different quality levels. A receiver initially subscribes to the lowest layer (base layer) and successively attempts to join the next higher layers to find its optimal subscription level. The subscription strategy is to only perform a join attempt if the receiver does not experience losses as congestion indication for a certain period of time. If loss is detected the receiver will unsubscribe from the currently highest layer. RLM's subscription management mechanism is based on detection of loss only and has been found to be unfair to TCP.

In an attempt to address the problems of RLM, Vicisano et al. developed the Receiver-driven Layered Congestion Control (RLC) [VRC98]. Periodic traffic bursts are employed for bandwidth inference and synchronization points are used to indicate to the receivers when they may join a higher layer. To address TCP-fairness and resemble TCP's AIMD behavior, the bandwidth of each layer is dimensioned such that the cumulative data rate increases exponentially. The time a receiver has to wait before trying to join the next layer also increases exponentially with the subscription level. As a result, the bandwidth increases proportionally to the time required to elapse before being allowed to join the next layer. On detection of packet loss, however, the receiver immediately unsubscribes from the highest layer. Thus, it reacts similar to TCP by halving its rate in case of losses.

Although RLM and RLC have been developed several years ago, they are still the most frequently cited approaches for layered multicast transmission of streaming video. Nevertheless, Legout and Biersack presented a study on pathological behaviors that both schemes might exhibit [LB00a]. The study indicates that under several conditions the mechanism of RLC also may cause unfair behavior. In order to address this issues, Legout and Biersack developed the Packet-pair Layered Multicast (PLM) scheme [LB00b], which is based on the generation of packet pairs for inference of the available bandwidth. This approach, however, requires all routers to implement fair queuing.

Layered Video Multicasting with Retransmissions (LVMR) is a system proposed by Li et al. [LPA98] that uses a hierarchy of agents in the network. These agents collect information from the receivers in the corresponding subtree and coordinate their group join and leave attempts. While this approach decreases the volume of control information compared to RLM, it introduces considerable overhead in terms of building the agent hierarchy.

So far, in all of the aforementioned multi-rate schemes the data rate of each layer is statically allocated. Dynamic layers in multi-rate congestion control have been introduced in Fine-Grained Layered Multicast with Dynamic Layers (FLID-DL) [BFH$^+$00, BHL$^+$02] by Byers et al. and Wave and Equation Based Rate Control (WEBRC) by Luby et al. [LGSH02]. In order to emulate rate increase and decrease in these schemes, the bandwidth consumption of each layer periodically decreases and increases. To maintain a constant rate, a receiver has to periodically join a certain number of layers. In FLID-DL a receiver's rate is reduced by the receiver simply not joining additional layers, and it is increased by joining multiple layers. Maintaining a constant rate in WEBRC requires a receiver to periodically join layers at a certain point of time. Both schemes utilize a significant number of multicast groups per session. They are closely coupled with routing and IGMP joins and leaves introducing significant overhead in multicast group maintenance.

In order to combine the potential advantages of both sender-driven and receiver-driven rate adaptation approaches, Sisalem and Wolisz [SW00] developed Multicast Loss-Delay based Adaptation (MLDA), a TCP-friendly congestion control framework for heterogeneous multicast. MLDA relies on receiver join and leave actions to control short-term network load according to the current congestion state. Similar to TFMCC, receivers measure their round trip-time and loss rate to calculate a TCP-fair share. In addition to using this estimate for subscription decisions, receivers are expected to report their estimate to the source for sender-side adaptation of the transmission rates. For feedback control MLDA employs partial suppression with which the possible range of reported values is divided into several intervals. For each interval, the well-known timer cancelation approach is applied, requiring each receiver to see the feedback messages of the other participants. Upon having collected the reports of a feedback round, the source evaluates the minimum and the maximum reported values. The transmission rates are then adjusted such that each layer increases the rate equally.

A parallel work to ours is the Hybrid Adaptation for Layered Multicast (HALM) proposed by Liu et al. [LLZ02]. It belongs to the class of hybrid sender- and receiver-driven schemes and builds on some of the concepts of MLDA. The distinct feature of HALM is optimal allocation of layer rates according to reported receiver bandwidth capabilities, which is very similar to our approach. However, while HALM applies network-centric optimization strategy, we contribute an application-aware optimization metric and a corresponding optimization algorithm. Furthermore, in HALM the feedback report period scales with the number of receivers while we employ a feedback scheme based on probabilistic sampling. Finally, in HALM join and leave attempts are solely based on the actual fair share estimate while our work proposes a timer-based strategy that leads to smoother subscription behavior.

The most recent hybrid approach is Smooth Multi-rate Multicast Congestion Control, the work of Kwon and Byers [KB03]. In SMCC layers are subject to dynamic adaptation within predetermined bounds for each layer, which implies a certain limit on the rate adaptation. In contrast to MLDA and HALM, each layer employs TFMCC as the underlying control mechanism so that the rate of each layer is dictated by the limiting receiver of the corresponding group. The main contribution of SMCC is the probing mechanism that tackles the possible inaccuracy of the fair share estimation algorithm. A receiver calculating a fair share that exceeds the next higher layer's rate does not immediately attempt to join. Instead it subscribes to a certain subset of probing layers first in order to emulate AIMD-like probing, and it decides to join the next layer only if the probing succeeded without experiencing a loss. The drawbacks of SMCC stem from the probing layers that introduce significant overhead and from the adaptation limits.

2.6.2 Stream Organization and Optimization

The objective of adaptive multi-rate multicast protocols has traditionally been the maximization of the aggregated bandwidth of a session. Shacham presented in [Sha92] an algorithm for allocating transmission rates such that the aggregated quality of all receivers is maximized. Thereby, receivers belonging to the same transmission group are assumed to have equal quality and fairness within the session. The latter aspect makes the metric not very suitable for optimization in heterogeneous environments. It tends to bias receivers with higher available bandwidth and it sacrifices receivers with relatively narrow capabilities.

Gorinsky and Vin adopted the above algorithm and evaluated the benefit of adaptive transmission rates based on a linear and an arbitrarily chosen convex utility function [GV01]. They showed that feedback-free allocation schemes might require by the order of two more multicast groups to reach a similar level of absolute distortion compared to the optimization-based scheme. However, while the authors performed an extensive set of simulations by varying different parameters, their work lacks a discussion of intra-session aspects.

Jiang et al. derived the inter-receiver fairness as an alternative metric for capturing the session performance considering intra-session aspects [JAZ98]. The inter-receiver fairness of a multicast session is defined as the weighted sum of the individual receiver fairness values. A receiver's individual fairness is thereby expressed in terms of a utility index that is directly related to the ratio of the receiver's achieved throughput to its fair share. The inter-receiver fairness is a popular representative of intra-session metrics for multi-rate multicast optimization purposes. It has been adopted by several other studies in recent years. In the original work the authors limited their discussion on the case of two multicast layers and derives a heuristic for partitioning the receivers into appropriate groups.

Yang et al. adopted the above inter-receiver fairness metric in order to calculate optimal receiver partitions for multicast sessions with an almost arbitrary number of groups [YKL00]. For this purpose the authors presented an optimization algorithm that has time complexity $O(N^3)$ and requires $O(N^2)$ auxiliary storage space. Liu et al. integrated a similar algorithm into an adaptation framework for layered multicast that adapts the transmission rates according to receiver feedback information [LLZ02]. Recently, Yousefi'zadeh et al. introduced extrapolation techniques to replace the non-continuously differentiable fairness function of individual receivers [YJ04]. Based on the extrapolated rational function they formulated the rate allocation and partitioning problem as a two-phase iterative process [YJH05]. The proposed algorithm provides a near-optimal solution with a time complexity $O(N \log N)$ and space complexity $O(N)$ when the number of iterations is low compared to N.

The underlying fairness function of the aforementioned solutions captures a measure of utility that is directly related to the ratio of a receiver's achieved throughput to its fair share. That is, the level of fairness a receiver is experiencing increases linearly with the transmission rate until reaching the maximum value at the receiver's fair share rate. While this definition maps from a local (bandwidth) to a global metric (fairness), it does neither take into account inter-session aspects nor application-specific characteristics. In particular, the level of user satisfaction in terms of perceived video quality is not reflected in that model, though the Internet was designed to meet the needs of users and their applications [She95].

2.6.3 Feedback Control

Scalable feedback mechanisms are crucial building blocks for most multicast applications and protocols. The classical examples for feedback usage are reliable multicast transport protocols that utilize feedback for positive acknowledgments (ACKs) and negative acknowledgments (NACKs) of packets. Due to the inherent feedback implosion problem [Dan89] and increasing interest in reliable multicast data delivery, substantial effort has been devoted to the development of feedback control schemes. Since in the Internet the fraction of lost packets is usually low compared to the fraction of received packets, NACK-based feedback schemes are considered more scalable than their ACK-based counterparts. Thus, the bulk of research in the area of reliable multicast has resort to NACK-based schemes and use timers for feedback suppression.

Floyd et al. in [FJL$^+$97] presented the Scalable Reliable Multicast Framework (SRM) that makes use of random timers. Repair requests are sent to the multicast group rather than a specific sender and any member of the group might schedule the transmission of a repair packet. The framework utilizes the concept of local recovery in a way that the closer a host is to the originator of the NACK the more probably it will send the repair. DeLucia and Obraczka proposed the use of selected representatives of a group [DO97]. These receivers are permitted to immediately send NACKs when experiencing a loss. All other receivers are supposed to start a feedback timer once detecting a loss and only send a NACK if no other host has already sent a request for that packet in the meantime. Grossglauser devised a scheme where feedback is unicasted to the source, and receivers are not required to listen to possible requests of others in [Gro97]. For that purpose the scheme relies on an algorithm that sets the feedback timers deterministically based on the proximity of receivers. Also Nonnenmacher and Biersack discussed the issue of parameterizing the feedback timer based on session and group characteristics [NB98]. They recommended the use of exponentially distributed timers that are scaled with increasing session size. Liang et al. use a similar approach but include the loss rate into the calculation of the timer [LHL00].

There are many other schemes (for example, [Hof96] and [PSLB97]) that use feedback suppression in order to enable the reliable transport of multicast data in a scalable way. On the other hand, there are few proposals that rely on aggregation rather than suppression. These approaches make use of network or overlay tree structures of multicast groups. Lehmann et al. in presented the Active Reliable Multicast (ARM) [LGT98] that relies on network support for caching of data and filtering of NACKs. Kasera et al. in their work [KHTK00] rely on router support similarly as Cain and Towsley proposing Generic Multicast Transport Services (GMTS) [CT00]. Chawathe et al. build an multicast overlay using Reliable Multicast proXies (RMX) [CMB00]. While the underlying mechanisms of these solutions might generally serve the needs of our work, they require placement of functionality into the network or at the edges.

The schemes discussed so far exploit redundancy of NACKs in order to control the amount of feedback. Some of the mechanisms might be used for window-based flow and congestion control, as for example in PGMCC proposed by Rizzo [Riz00]. For rate-based control of the transmission rate, however, mechanisms discriminating between different feedback values are necessary.

Bolot et al. presented a scalable feedback method that combines a probabilistic polling mechanism with increasing search scope and a randomly delayed reply scheme [BTW94]. This scheme is used in the INRIA Videoconferencing System (IVS) to estimate the group size and to detect the network state corresponding to the worst positioned receiver. The probabilistic mechanism relies on random keys generated by the source and the receivers. When soliciting feedback, the source sends out a feedback requests containing the current network state information and the number of significant digits of the random key. A receiver only responds if it matches the key and it perceives the network worse than the current advertised state from the sender. Feedback is suppressed by initializing the number of significant digits to a reasonable low value and increasing it in subsequent feedback rounds. The scheme is not directly applicable for estimation of the bandwidth capability distribution of the active receivers.

A novel value-based feedback scheme that modifies the well-known concept of exponential timer-based feedback suppression has been presented by Widmer and Fuhrmann [WF01]. It is utilized in a sender-based multicast rate adaptation scheme [WH01] for detecting the limiting receiver of a multicast group. When a feedback request is solicited to the group, a receiver responds to the request only if its feedback timer is not canceled by feedback messages of other receivers. Biasing feedback timers in favor of receivers with a higher or lower value of the target metric is the core contribution of this approach. Nevertheless, it requires the distribution of receiver feedback responses to all other receivers introducing additional signaling overhead. Similar to the preceding approach, it is also not applicable for sampling the receiver capabilities.

Recently, research in the area of application-layer overlay networks has been intensified. These architectures overcome the requirement of placing agents or active routers in the network or at the edges. Chu et al. developed the End System Multicast (ESM) [CRSZ01] that has been used for live streaming of the ACM SIGCOMM 2002 conference. There are many more application-layer overlays, for example, HyperCast [LB99], Overcast [JGJ$^+$00], Scattercast [Cha03], and those presented in [BBK02] [UKB02], to name a few. Most of them include tree-building mechanisms that might be utilized to build feedback control overlays. However, major issues of application-layer overlay networks are the dynamic tree management and the implied complexity. While for small groups of tens or hundreds of receivers overlay systems provide a very interesting alternative to native multicast, scaling sessions to thousands and more end hosts still remains a challenge.

Although to our best knowledge there is currently no feedback scheme that solves the issue of our work, the developed probabilistic sampling mechanism has been partially inspired by the probabilistic suppression scheme of Bolot et al. [BTW94].

Chapter 3

Stream Organization

In this chapter, we focus on the development of an algorithm for optimizing the stream organization of a multicast session at the source. The objective of the algorithm is to calculate the set of operational rates that maximizes an intra-session performance index for a given number of multicast channels.

For that purpose, we first derive a model for expressing the utility and fairness of a receiver participating in a multicast session. Basically, the model relies on the usage of utility functions to map from the actual transmission rate to a user satisfaction scale. The utility value of a receiver is then transformed into an individual fairness index considering the bandwidth constrains of the corresponding receiver[1].

Subsequently, we define an intra-session performance metric that captures the average receiver utility fairness index and develop an algorithm for its optimization. Given a predefined number of multicast channels, the algorithm calculates the optimal set of transmission rates according to the distribution of the receiver bandwidth capabilities. The underlying fairness model and the optimization algorithm are not limited to a particular utility function but they are rather applicable to a general class of utility functions.

Based on objective video quality measures and an existing rate-distortion model for FGS-encoded video (see Section 2.5.1) we derive two application-aware utility functions. Considering standard test video sequences, we use the utility models and the developed optimization algorithm for quantitatively evaluating the impact of

---

[1] The bandwidth capability of a receiver is generally determined by constrained end device capabilities (e.g., processing power) and/or a network bottleneck (e.g., link capacity). Without the loss of generality, we focus on the case where only network elements limit the bandwidth capability to a fair share value, and synonymously use the term expected and available bandwidth.

the number of multicast groups and the benefit of the optimal group rate allocation strategy.

The roadmap of this chapter is summarized in Figure 3.1.

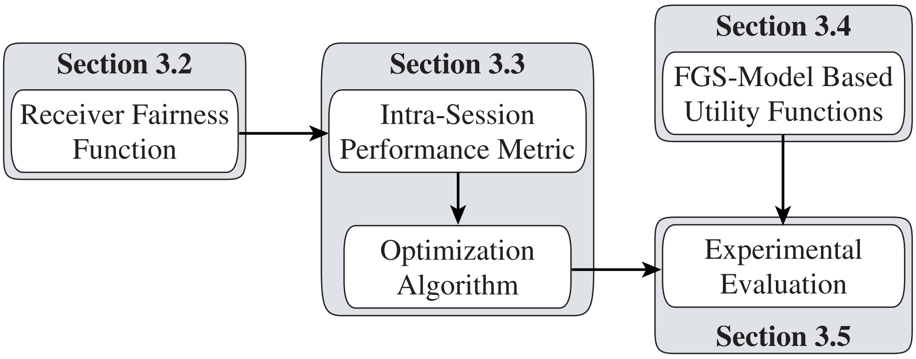

Section 3.2
Receiver Fairness Function
Section 3.3
Intra-Session Performance Metric
Optimization Algorithm
Section 3.4
FGS-Model Based Utility Functions
Experimental Evaluation
Section 3.5

Figure 3.1: Roadmap of the chapter on *Stream Optimization*.

3.1 Motivation

Native multicast is a very network-efficient transmission mode for homogeneous environments, especially when end devices have similar capabilities and face similar network conditions. Traditional single-rate schemes, however, cannot accommodate heterogeneous conditions. Rubenstein et al. [RKT02] analytically showed that multi-rate multicast sessions can achieve several desirable fairness properties that cannot be obtained in general networks by single-rate sessions. In theory maximum satisfaction and fairness are achieved when the number of sender-provided data rates equals the number of distinguishable bandwidth conditions of a session. In a heterogeneous environment such as the Internet, a corresponding transmission scheme requires the establishment and maintenance of numerous multicast groups per session. This implies substantial management cost regarding routing state maintenance and signaling overhead, which can contradict the benefits gained from the improved adaptation granularity. Last but not least, the allocation of each additional group increases packetization overhead. From an operational point of view a multi-rate scheme should obviously operate with only few multicast channels per session.

Following common practice and statically allocating only a small number of operational rates may severely degrade the performance of a multi-rate multicast session. Important factors for such degradation are heterogeneous transmission conditions and the distribution of receiver capabilities that are generally not known in advance, and are quite likely subject to more or less pronounced dynamics. A promising alternative to the static strategy is the adaptation of transmission rates to the actual conditions. Applying this strategy might significantly improve network utilization as well as collective user satisfaction without increasing network management costs, as we show by means of experiments. However, this approach demands for the definition of a reasonable optimization metric and an efficient rate allocation[2] algorithm at the server.

3.2 Modeling Receiver Utility Fairness

In the following section, we develop a metric for capturing the fairness value of a receiver participating in a multicast session. In the modeling process inter-session, intra-session, and application-specific aspects are considered. Inter-session fairness is applied to address the fact that the Internet is based on cooperation. More specifically, none of the flows should allocate more bandwidth than its fair share. Intra-session aspects are captured through a function of a receiver's operating rate and its expected bandwidth. Finally, application-specific aspects are modeled using a user utility function, which maps from the bit rate of a video to a user satisfaction value (see Section 2.5.2). We refer to the resulting metric as *receiver utility fairness*.

3.2.1 Inter-Session Fairness

Fairness is considered a very important issue for environments such as the Internet, where resources have to be shared cooperatively [JCH84, BM01]. Although multicast fairness has been extensively studied and discussed within the last few years [WS98, JZA99, GGHS99, ST00, Den00, LNB01, RKT02], a general consensus is still lacking on the relative fairness between multicast and unicast traffic. Basically, bandwidth might be allocated according to different policies [LNB01], for example, as a function of participating receivers. A multicast session might then be given more bandwidth than a TCP connection because it serves several receivers. On the other hand, this will penalize or even starve a TCP connection if it is sharing a common link with a multicast distribution tree to a large receiver set. Thus,

---

[2] We use the term rate allocation to refer to the process of partitioning the streaming data into a number of multicast channels at the source.

it is also reasonable not to allocate more bandwidth to a multicast session than to a TCP connection; the more so as TCP currently still makes 90 % and more of the Internet traffic.

The controversy about the above issue is still ongoing. In order for our model to be generally valid, we follow an abstract approach without the need to decide for any of the above paradigms. We assume the fair bandwidth share of a receiver to be determined by the underlying multicast fairness definition. This might be, for example, a multiple of a TCP connection or a TCP-compatible rate as defined in [BCC+98].

As already stated in the introductory part of this dissertation, we assume the Internet to provide only a best-effort service. End systems are expected to adopt the "social" rules and behave cooperatively in order to contribute to the stability of the Internet. Few flows sharing a bottleneck might experience degradation of QoS, such as increased loss rate or delay, even if one or more flows are only slightly misbehaving. Similar misbehavior in a environment of high statistical multiplexing might not necessarily lead to a severely degraded QoS experience, neither for the greedy flow nor the others. For example, if a single flow is aggressive and gets more bandwidth allocated than its fair share, the effect can usually be absorbed by the body of all sessions sharing the bottleneck. In that case, it might be negligible. However, the more flows apply this aggressive strategy, the higher is the impact on the experienced QoS and finally on the overall stability.

In the preceding paragraph we took a very simple reflection of a rather complex topic, which is subject to many research efforts. It is out of the scope of our work to contribute to this research area; consequently, in our model we apply a rather conservative policy and assume that a flow should not be allocated more than its fair bandwidth share. The detailed determination of this value will be discussed in Chapter 5.

3.2.2 Receiver Fairness Function

A widely-used receiver fairness function that maps from the actual operating rate to a fairness value has been originally defined by Jiang et al. [JAZ98]. This metric captures the individual fairness $f_i(r)$ of a receiver i as a linear function of its operating rate r, that is, its actual received rate, and the receiver's fair share r_i^* in the range $0 \leq r \leq r_i^*$. For $r > r_i^*$ the fairness decreases as data is lost. Thereby, a receiver is allowed to specify a maximum acceptable loss tolerance ξ_i, indicating the maximum fraction of transmitted data that can acceptably be lost:

$$f_i(r) = \begin{cases} \frac{\min(r_i^*,r)}{\max(r_i^*,r)} & \text{if} \quad 0 \le r \le \frac{r_i^*}{1-\xi_i}. \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

The function in Equation 3.1 captures a measure of utility that is directly related to the ratio of a receiver's achieved throughput to its fair share. That is, the level of fairness a receiver is experiencing increases linearly with the transmission rate until reaching the maximum value at the receiver's fair share rate. While this definition maps from a local (bandwidth) to a global metric (fairness), it does neither take into account the inter-session aspects elaborated in Section 3.2.1 nor application-specific characteristics. In particular, the level of user satisfaction in terms of perceived video quality is not reflected in the model.

However, since the Internet was designed to meet the needs of users and their applications, network performance must not be measured in terms of network-centric quantities but rather in terms of the degree to which it satisfies the service requirements of user applications [She95]. For that purpose, user utility functions based on subjective assessment or mathematical quality measurements such as the PSNR and MSE can be used (see Section 2.5).

Receiver Utility Fairness

To overcome the aforementioned limitations, we extend the original metric to access the receiver fairness as a general function of the receiver utility u, which we refer to as the *receiver utility fairness* throughout this thesis:

$$f_i(r) = \begin{cases} \frac{u(r)}{u(\min\{r_i^*, r_{max}\})} = \frac{u(r)}{u_i} & \text{if} \quad 0 \le r \le \min\{r_i^*, r_{max}\}. \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

In Equation 3.2 the fairness function is generalized to the ratio of the utility $u(r)$ experienced at the operational rate r to the utility $u_i = u(c_i)$ a receiver i would achieve when the transmission rate equals its bandwidth capability c_i. We define the latter as $c_i = \min\{r_i^*, r_{max}\}$, where r_i^* denotes the receiver's optimal operational rate (i.e., its fair share) and r_{max} is the highest possible rate provided by the server.

The receiver utility fairness function is a wide-sense increasing function that satisfies the following constrains:

1. $f_i(r) = f_j(r) \quad \text{if} \quad c_i = c_j$.
   The utility function is transferable making receivers' fairness indices comparable and summable.

2. $f_i(r) \in [f_{min}, f_{max}]$.
   The fairness value is normalized to a quantity between $f_{min} = 0$ and $f_{max} = 1$.

3. $f_i(r) = f_{max} \quad \text{if} \quad r = c_i$.
   The fairness value is maximized when the operational rate r equals a receiver's bandwidth capability c_i.

4. $f_i(r) < f_{max} \quad \text{if} \quad r \neq c_i$.
   The function has only a single maximum.

5. $\frac{df_i(r)}{dr} \geq 0 \quad \text{if} \quad 0 \leq r < c_i$.
   The fairness function is non-decreasing in the interval $[0, c_i)$.

Note that, in accordance to the inter-session considerations elaborated in Section 3.2.1, the ratio of the utility values in Equation 3.2 is constrained to values $r \leq r_i^*$ if $r_i^* \leq r_{max}$. This matches the common practice since multicast congestion control generally relies on receiver-driven group join and leave decisions. That is, if the transmission rate exceeds the fair share of a receiver, the congestion indication should force the receiver to leave the corresponding group.

While incorporating the original aspect of inter-receiver fairness, our model furthermore allows for an easy integration of utility functions that map the rate of the delivered video to an application-aware performance measure. The latter is usually obtained from rate-distortion characteristics or perceptual video measures in the context of video transmission, as discussed in Section 2.5.

3.3 Optimizing Group Rates

The derived metric for capturing the fairness of individual receivers participating in a multicast session provides means for intra-session performance evaluation and optimization. Thus, in the following section we introduce some basic terms and derive an intra-session performance metric to continue with the formulation of the optimization problem and development of a corresponding algorithm.

3.3.1 Intra-Session Performance Metric

Suppose that there are N destinations in a multi-rate multicast session having fair shares $r_1^*, \ldots, r_N^*$ and corresponding bandwidth capabilities $c_1, \ldots, c_N$. Furthermore, the streaming server is expected to transmit the video at a given and limited number L of different quality levels q_l. In the following, we assume that the quality levels are ordered such that a higher index l indicates a higher quality level.

We define $\mathbf{g}_{1,L} = (g_1, \ldots, g_L)$ to denote an L-tuple of distinct receiver group rates that correspond to the different quality levels. Since normally a higher quality level implies a higher data rate, for the components of $\mathbf{g}_{1,L}$ it follows that $g_i < g_j$ if $i < j$.

Note that the term receiver group rate is used to denote the receiving rate of the group of destinations that is receiving the content at the same quality level. In the case of layered transmission, a receiver of group l is subscribed to the multicast channels 1 through l, and g_l equals the sum of the transmission rates of all subscribed multicast channels. In the case of replicated streams, a receiver is subscribed to a single multicast channel only so that g_l equals the transmission rate of that particular channel.

Since the local objective of each receiver is to maximize its utility under the inter-session constrains, a receiver i will normally choose to become member of the receiver group l that provides the highest possible group rate g_l considering the receiver's bandwidth capability c_i.

As a consequence, for the actual operational rate r_i of the receiver i it follows:

$$r_i = r_i(\mathbf{g}_{1,L}) = \max\{g_l : \; g_l \leq c_i\}. \tag{3.3}$$

Figure 3.2 illustrates the interrelation of the bandwidth capabilities and the group rates.

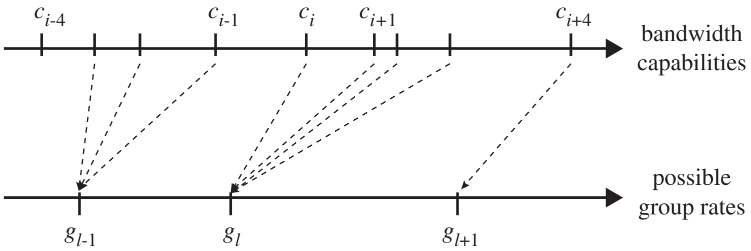


Figure 3.2: Schematic illustration of the interrelation of the receiver bandwidth capabilities and the group rates.

Intra-Session Utility Fairness

By generalizing the definition of the inter-receiver fairness from [JZA99], we define the *intra-session utility fairness* to capture the intra-session performance in terms of the average value of the individual receiver utility fairness values:

$$U = \frac{1}{N} \sum_{i=1}^{N} f_i(r_i). \tag{3.4}$$

Using Equation 3.3, the intra-session utility fairness can then be expressed as a function of the vector $\mathbf{g}_{1,L}$ of allocated group rates:

$$U(\mathbf{g}_{1,L}) = \frac{1}{N} \sum_{i=1}^{N} f_i(g_l). \tag{3.5}$$

It gives a measure of how fairly the receivers of the same multi-rate multicast session are served by the provided receiver group rates considering user utility.

3.3.2 Optimization Algorithm

Since the Internet was designed to meet the needs of users, a network service should be measured in terms of an application-aware user utility [She95]. We follow this requirement in contrast to most of the existing approaches that neglect this fundamental requirement and perform optimization based on network-centric quantities such as the bandwidth. Consequently, the optimization objective of our work can be described as follows:

> *Partition the video data into a given number of distinct multicast groups such that the resulting tuple of allocated group transmission rates maximizes the intra-session utility fairness.*

Let $\mathbf{G}_{1,L}$ denote the set of possible L-tuples of group rates. Using Equation 3.3 and Equation 3.5 the optimization problem is then formally stated as:

$$\begin{aligned} &\underset{\mathbf{g}_{1,L} \in \mathbf{G}_{1,L}}{\text{maximize}} && U(\mathbf{g}_{1,L}) \\ &\text{subject to} && \mathbf{G}_{1,L} = \{\mathbf{g}_{1,L} : g_{l-1} < g_l \, \forall l \in \{1, \dots, L\}\}, \end{aligned} \tag{3.6}$$

and can be solved by an iterative procedure, as we present subsequently.

Let $\mathbf{g}_{l,m} = (g_l, \dots, g_{l+m-1})$ denote an m-tuple of consecutive components of the group rate vector $\mathbf{g}_{1,L}$ and $\mathbf{G}_{l,m} = \{\mathbf{g}_{l,m} : g_{k-1} < g_k \, \forall k \in \{l+1, \dots, m\}\}$ the corresponding set of possible m-tuples. We then define the aggregate utility fairness F of an m-tuple as follows:

$$F(\mathbf{g}_{l,m}) = \sum_{g_l \le c_i < g_{l+1}} f_i(g_l) \; + \ldots + \sum_{g_{l+m-1} \le c_i < g_{l+m}} f_i(g_{l+m-1}), \tag{3.7}$$

and the aggregate utility fairness of the lth receiver group as:

$$F_1(g_l) = F(\mathbf{g}_{l,1}) = \sum_{g_l \le c_i < g_{l+1}} f_i(g_l). \tag{3.8}$$

Applying the above notation to Equation 3.5 and assuming $g_{L+1} > g_L$ and $g_{L+1} > c_i$ for all $i \in \{1, \ldots, N\}$, we can rewrite the intra-session utility fairness as:

$$U(\mathbf{g}_{1,L}) = \frac{1}{N} F(\mathbf{g}_{1,L}), \tag{3.9}$$

Obviously, to maximize $U(\mathbf{g}_{1,L})$ is equivalent to maximizing $F(\mathbf{g}_{1,L})$, which leads to the following recursive expression:

$$\begin{aligned}
\hat{F}(\mathbf{g}_{1,L}) &= \max_{\mathbf{g}_{1,L} \in \mathbf{G}_{1,L}} F(\mathbf{g}_{1,L}) \\
&= \max_{\mathbf{g}_{1,L} \in \mathbf{G}_{1,L}} \{F_1(g_1) + \ldots + F_1(g_L)\} \\
&= \max_{g_L} \left\{ \max_{\mathbf{g}_{1,L-1} \in \mathbf{G}_{1,L-1}} \{F_1(g_1) + \ldots + F_1(g_{L-1})\} + F_1(g_L) \right\} \\
&= \max_{g_L} \left\{ \hat{F}(\mathbf{g}_{1,L-1}) + F_1(g_L) \right\},
\end{aligned} \tag{3.10}$$

and can be solved by means of dynamic programming to find the optimal group rate vector.

Reducing Complexity

In practical terms, the possible values that can be assigned to the L group rates $g_1, \ldots, g_L$ generally depend on the underlying encoding of the content. FGS-encoded video can be truncated and striped at a bit-level granularity allowing the selection of almost any arbitrary bandwidth value. This comes with an increased computational complexity: for a multicast session of L channels and M possible rate allocation levels there are $\binom{M}{L}$ possible combinations for choosing the group rates.

However, considering some characteristics of the FGS codec and the general constrains of the utility function allows us to reduce the complexity in order to make the optimization problem more tractable:

1. Recall that scalable codecs usually provide a base quality with a predetermined rate r_{base}, and limit the transmission rate to the rate r_{max} that corresponds to the highest available quality level. Thus, the rate of the first receiver group equals the base layer $g_1 = r_{base}$, and the transmission rate of the highest order group cannot exceed the maximum rate $g_L \leq r_{max}$.
2. From properties 3–5 of the receiver utility fairness discussed in Section 3.2.2 and from Equation 3.8 it follows that the aggregate fairness function $F_1(g_l)$ of group l has only discontinuity points for $g_l = c_i$. Furthermore, it is a piecewise wide-sense increasing function in the intervals $I_i = [c_i, c_{i+1})$:

   $$\frac{d}{dg_l} F_1(g_l) = \sum_{g_l \leq c_i < g_{l+1}} \frac{d}{dg_l} f_i(g_l) \geq 0 \quad \forall g_l \in I_i. \tag{3.11}$$

   This implies that it is sufficient to test for the endpoints of the intervals I_i in order to find the optimal set of group rates to maximize the overall intra-session performance.

The optimization problem can be finally reformulated as:

$$\begin{aligned} \underset{\mathbf{g}_{1,L} \in \mathbf{G}_{1,L}}{\text{maximize}} \quad & F(\mathbf{g}_{1,L}) \\ \text{subject to} \quad & \mathbf{G}_{1,L} = \{\mathbf{g}_{1,L} : g_{l-1} < g_l \, \forall l \in \{2, \ldots, L\}\}, \\ & g_1 = r_{base}, \\ & g_L \leq r_{max} < g_{L+1}, \\ & g_l \in \{c_1, \ldots, c_N\} \, \forall l \in \{1, \ldots, L\}, \end{aligned} \tag{3.12}$$

and can be solved by applying a dynamic programming algorithm similar to [YKL00]. The corresponding algorithm has time complexity and auxiliary storage space requirements $O(N^2)$ for the precomputation of the aggregate group fairness indices. The calculation of the optimal receiver groups implies $O(N^2)$ time complexity.

However, we have found a more efficient algorithm by exploiting the characteristics of the underlying fairness function for the precomputation. Expressing the receiver utility fairness function $f_i(g_l) = \frac{u(g_l)}{u(c_i)}$ in Equation 3.8 and rearranging it leads to the following formula:

$$
\begin{aligned}
F_1(g_l) &= \sum_{g_l \le c_i < g_{l+1}} \frac{u(g_l)}{u(c_i)} \\
&= u(g_l) \sum_{g_l \le c_i < g_{l+1}} \frac{1}{u(c_i)} \\
&= u(g_l) \left(\sum_{c_i \ge g_l} \frac{1}{u(c_i)} - \sum_{c_i \ge g_{l+1}} \frac{1}{u(c_i)} \right). \qquad (3.13)
\end{aligned}
$$

In the above form the precomputation requires only the iterative summation of the inverses of the receiver utilities reducing both time complexity and auxiliary storage space to $O(N)$, as shown in Algorithm 3.1.

3.4 Deriving Utility Functions

In order to perform quantitative performance analysis based on the derived utility fairness function and the developed optimization algorithm, we present reasonable utility functions in this section.

The use of objective and state-of-the-art measures for accessing the distortion and quality of a video signal, namely the MSE and PSNR, has been introduced in Section 2.5.3. In the following, we first derive the general receiver utility function $u(r)$, and subsequently map the values obtained by the mathematical measurements to a utility value expressed in terms of a Mean Opinion Score (MOS) scale ranging from 1 (best) to 5 (worst) according to [ITU02].

Scalable video codecs are usually parameterized such that the base layer is encoded in order to provide a minimum acceptable quality while still supporting receivers with low bandwidth capabilities. Consequently, we map the quality at the base rate r_{base} to the minimum utility value $u_{min} = u(r_{base}) = 1$. Furthermore, the maximum utility is achieved when there is no impairment between the original video and the reconstructed video. This is the case when the video is received with the maximum possible rate r_{max} provided by the server, so that $u_{max} = u(r_{max}) = 5$. We then formally state the utility function as:

$$
u(r) = u_{min} + \int_{r_{base}}^{r} \frac{d}{dx} u(x) dx \qquad \text{if } r_{base} \le r \le r_{max}. \qquad (3.14)
$$

```
Input:
  $N$          – number of receivers
  $L$          – number of receiver groups
  $c_i$        – bandwidth capability of the $i$th receiver.
  $r_{base}$   – data rate of the base stream
Output:
  $\hat{F}_l[i]$  – maximum aggregate fairness for receivers 1 to $i$ and $l$ groups
  $s[i]$          – sum of reciprocals of the utilities of receivers $i$ to $N$
  $k_l[i]$        – index of the first receiver of the $l$th group for $\hat{F}_l[i]$
  $g_l$           – optimal transmission rate for the $l$th group

// Precompute sum of reciprocals of receiver utilities
$s[N+1] \leftarrow 0$
for $i \leftarrow N$ to 1 do  $s[i] \leftarrow s[i+1] + 1/u(c_i)$

// Calculate optimal receiver groups
for $i \leftarrow 1$ to $N$ do  $\hat{F}_1[i] \leftarrow u(r_{base}) \cdot (s[1] - s[i])$
for $l \leftarrow 2$ to $L$ do
    for $i \leftarrow 1$ to $N$ do
        $\hat{F}_l[i] \leftarrow \hat{F}_{l-1}[i]$
        $k_l[i] \leftarrow i + 1$
        for $j \leftarrow 1$ to $i - 1$ do
            $F_{tmp} \leftarrow \hat{F}_{l-1}[j] + u(c_{j+1}) \cdot (s[j+1] - s[i])$
            if $F_{tmp} > \hat{F}_l[i]$ then
                $\hat{F}_l[i] \leftarrow F_{tmp}$
                $k_l[i] \leftarrow j + 1$
            end
        end
    end
end

// Assign optimal group rates
$g_1 \leftarrow r_{base}$
$i \leftarrow N$
for $l \leftarrow L$ to 2 do
    $k \leftarrow k_l[i]$
    $g_l \leftarrow c_k$
    $i \leftarrow k - 1$
end
```

Algorithm 3.1: Pseudo-code for optimal rate allocation algorithm.

As previously discussed, a quality metric $q(r)$ that is assumed to correlate with the user utility exhibits similar marginal gain in the allowed range $r_{base} \leq r \leq r_{max}$:

$$\frac{d}{dr}u(r) = \eta_q \frac{d}{dr}q(r), \tag{3.15}$$

where η_q denotes a normalization factor that can be straight-forwardly derived from Equation 3.14 for r_{max}, which leads to:

$$u(r) = u_{min} + \frac{u_{max} - u_{min}}{q(r_{max}) - q(r_{base})}\left(q(r) - q(r_{base})\right). \tag{3.16}$$

For the purpose of this work we derive two instantiations of the utility function based on the PSNR and MSE as previously motivated:

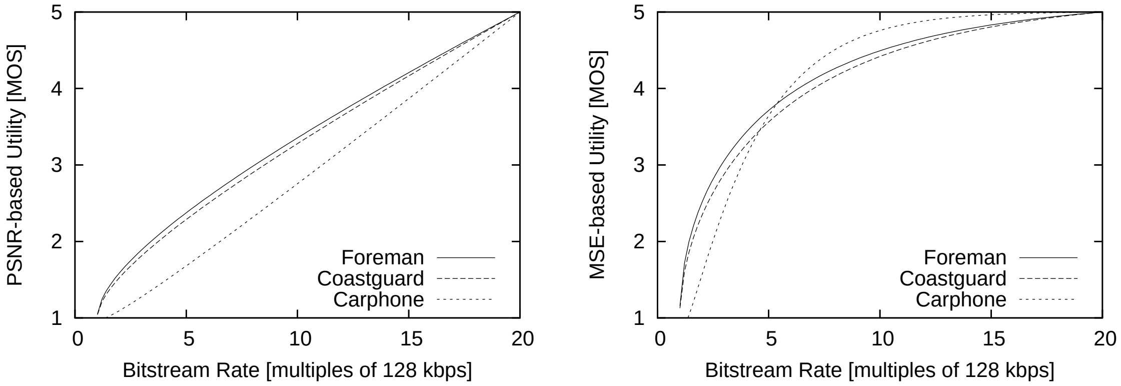
$$\frac{d}{dr}q(r) = \frac{d}{dr}\mathrm{PSNR}(r) \quad \Rightarrow \quad u_{\mathrm{PSNR}}(r) = 1 + 4\frac{\mathrm{PSNR}(r) - \mathrm{PSNR}(r_{base})}{\mathrm{PSNR}(r_{max}) - \mathrm{PSNR}(r_{base})}, \tag{3.17}$$

$$\frac{d}{dr}q(r) = -\frac{d}{dr}\mathrm{MSE}(r) \quad \Rightarrow \quad u_{\mathrm{MSE}}(r) = 1 + 4\frac{\mathrm{MSE}(r_{base}) - \mathrm{MSE}(r)}{\mathrm{MSE}(r_{base}) - \mathrm{MSE}(r_{max})}. \tag{3.18}$$

3.5 Experimental Evaluation

In this section, we examine the performance of our group rate allocation scheme for MPEG-4 FGS-encoded video sequences under a variety of group characteristics. We study the achievable performance and compare it with traditional static schemes using instantiations of the utility function from Equation 3.17 and 3.18 with the square-root model from Equation 2.8 and Equation 2.9, respectively.

All experiments have been conducted with the parameters for three different video test sequences—Foreman, Coastguard, Earphone—used in the standardization process of MPEG-4. The coding parameters and the parameters of the square-root model are listed in Appendix A. The base layer is encoded with $r_{base} = 128\,\mathrm{kbps}$, and the maximum transmission rate is set as high as $r_{max} = 20 \cdot r_{base} = 2,560\,\mathrm{kbps}$ to accommodate a high level of heterogeneity regarding receiver bandwidth capabilities. Figure 3.3 shows the resulting utility curves for the three video sequences. Since the results for all three video sequences proved to be very similar, in the remainder of the work we only illustrate the results of the experiments performed for the Foreman video.



(a) PSNR-based utility function u_{PSNR} (b) MSE-based utility function u_{MSE}

Figure 3.3: Receiver utility fairness function for different video sequences encoded with $r_{base} = 128\,\mathrm{kbps}$ and $r_{max} = 2,560\,\mathrm{kbps}$.

For each experiment we calculated the mean value of 30 independent experiments, and in order to reveal the significance of the results we calculated the two-sided 95 % confidence intervals. Each experiment is based on sets of $N = 1,000$ receivers with capabilities directly modeled as coming from different distributions (see Appendix B):

Uniform. Receiver capabilities are evenly distributed over the interval $[r_{base}, r_{max}]$. This is basically the most difficult distribution regarding optimization, since there is no distinct mode where capabilities are clustered around. Thus, it is considered the worst-case scenario in our experiments and the intra-session performance is expected to be lowest compared to all other distributions.

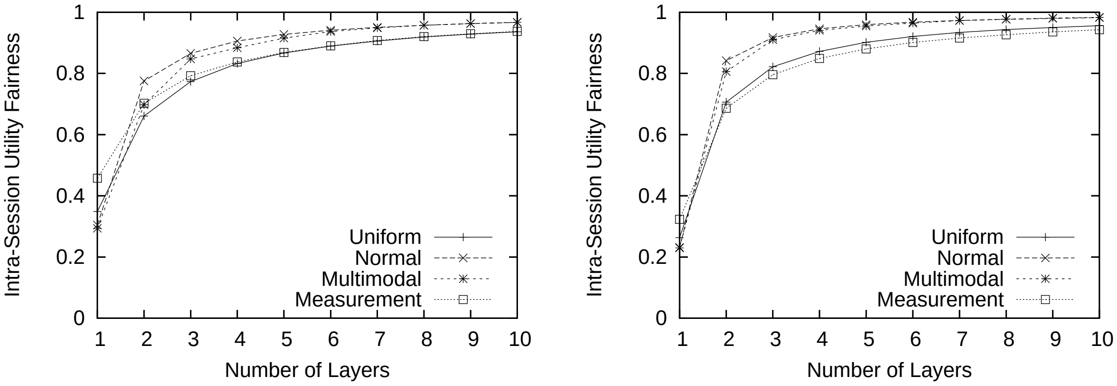
Normal. The receiver capabilities are normally distributed $N(\frac{r_{max}-r_{base}}{2}, \frac{r_{max}}{8})$. Receiver capabilities are clustered around the mean and are relatively homogeneous. As a consequence, this is a rather optimistic configuration and expected to provide comparably high intra-session performance for even a very small number of multicast groups.

Multi-modal. The receiver capabilities are assumed to belong to three different clusters with distinct modes. Each cluster is bounded on the left side by the base rate r_{base}, and on the right side by its mode. The capabilities of each cluster are beta distributed $B(7, 1)$ such that the density is the highest close to the corresponding mode. We assume the following three modes and their proportion of the total number of receivers: 1 Mbps 40 %, 2 Mbps 40 %, and 3 Mbps 20 %.

Measurement-based. While the former distribution are based on theoretical models, the last distribution we use is based on data measured over the Internet as explained in Appendix B.5. We thus consider it a more realistic distribution for large and widely distributed sessions.

3.5.1 Impact of the Number of Transmission Rates

In a first experiment we study the performance of the optimal allocation scheme with respect to the available number of multicast groups. Figure 3.4(a) and Figure 3.4(b) show the resulting mean values of the intra-session utility fairness for the PSNR-based respectively MSE-based utility function over the number of groups. Note that we did not include the confidence intervals since they proved to be very small, which confirms the statistical significance of the results.



(a) PSNR-based utility function u_{PSNR} (b) MSE-based utility function u_{MSE}

Figure 3.4: Achievable intra-session utility fairness with the adaptive allocation strategy for different receiver capability distributions.

We generally observe that there is a significant increase in intra-session performance (intra-session utility fairness) for up to four quality levels when group rates are allocated according to the optimal strategy utilizing our algorithm. For all tested distributions and both utility functions four multicast groups already provide intra-session performance of over 80 %. Compared to the single-rate case, four layers increase the intra-session utility fairness index by over 25 % for the PSNR-based metric and over 40 % for the MSE-based metric. The gain for each additional layer is marginal so that the relative improvement for ten groups compared to four groups is approximately within 10 %. Since the implied costs for establishing and maintaining additional multicast groups are increasing much faster, we consider the allocation of four rate-adaptive multicast groups to provide a reasonable compromise. This observation holds for possible transmission rates in the interval $[r_{base}, 20 \cdot r_{base}]$. For a constant number of multicast groups and increasing interval size, the intra-session performance will decrease; to reach a comparable performance level the number of allocated groups would have to be increased. In similar experiments for a range of the transmission rate of $[r_{base}, 40 \cdot r_{base}]$, we obtained comparable results for the intra-session utility fairness when five respectively six groups were allocated.

Figure 3.4 also demonstrates that the optimization scheme provides better results for the normal and multi-modal distributed sets when compared to the uniform and measurement-based distributed sets. Both of the former have both modes with clustered capabilities so that a single multicast group allocated close to a mode will significantly increase the aggregate utility fairness of that group. In contrast, the latter two distributions do not provide distinct clusters or modes.

3.5.2 Comparison with Static Allocation Strategies

In the next experiments, we compared the optimal allocation strategy to traditional additive and multiplicative strategies.

Schemes using an additive strategy organize their provided streams such that group transmission rate is increased by a constant value for each subsequent group:

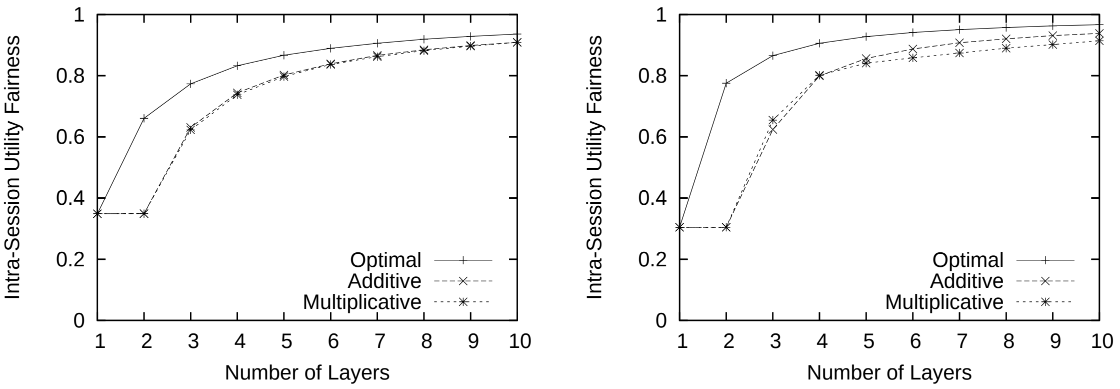
$$g_l = r_{base} + \frac{l-1}{L-1}\left(r_{max} - r_{base}\right) \qquad \text{for} \quad 1 \le l \le L,\ L > 1. \tag{3.19}$$

For the multiplicative strategy, streams are organized such that the transmission rate of each subsequent group is scaled by a constant factor:

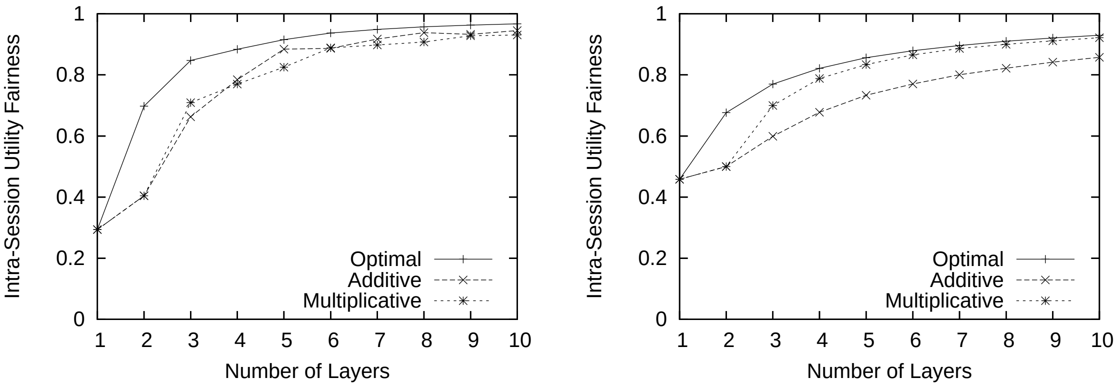
$$g_l = r_{base}\left(\frac{r_{max}}{r_{base}}\right)^{\frac{l-1}{L-1}} \qquad \text{for} \quad 1 \le l \le L,\ L > 1. \tag{3.20}$$

The results of the experiments for different receiver capability distributions are illustrated in Figure 3.5 for the PSNR-based utility function and in Figure 3.6 for the MSE-based utility function.

We observe that for a small number of groups the optimal allocation strategy



(a) uniform distribution (b) normal distribution

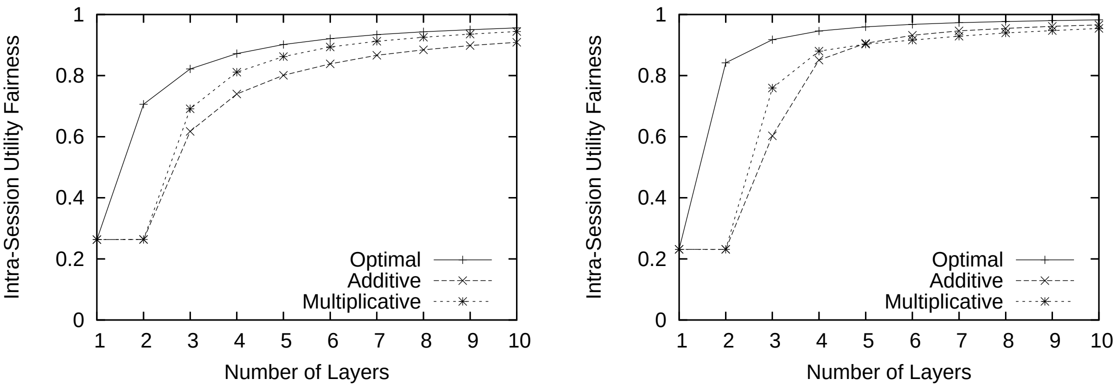


(c) multimodal distribution (d) measurement-based distribution

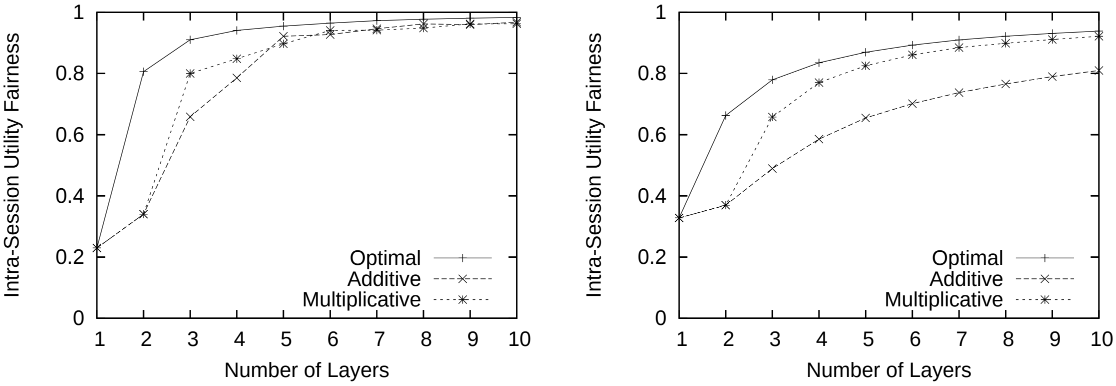
Figure 3.5: Performance in terms of PSNR-based intra-session utility fairness achieved for different rate allocation strategies and distributions of receiver bandwidth capabilities.

outperforms the adaptive as well as the multiplicative strategy. The difference when allocating four layers ranges from 12 % to almost 20 % in the case of the PSNR-based utility function for the uniform, normal and multi-modal distributed capability sets (see Figure 3.5(a)–3.5(d)). While for the measurement-based sets the optimal scheme clearly outperforms the additive strategy by similar values, the difference to the multiplicative scheme is around 4 % (see Figure 3.5(d)).

Due to the pronounced concave characteristics of the MSE-based utility function, the marginal utility only slightly increases for higher transmission rates (see Figure 3.6). Hence, the optimal strategy applied to four groups outperforms the other strategies by 7–17 %, which are slightly lower values compared to the PSNR-based results. For the measurement-based distribution and additive strategy how-



(a) uniform distribution (b) normal distribution



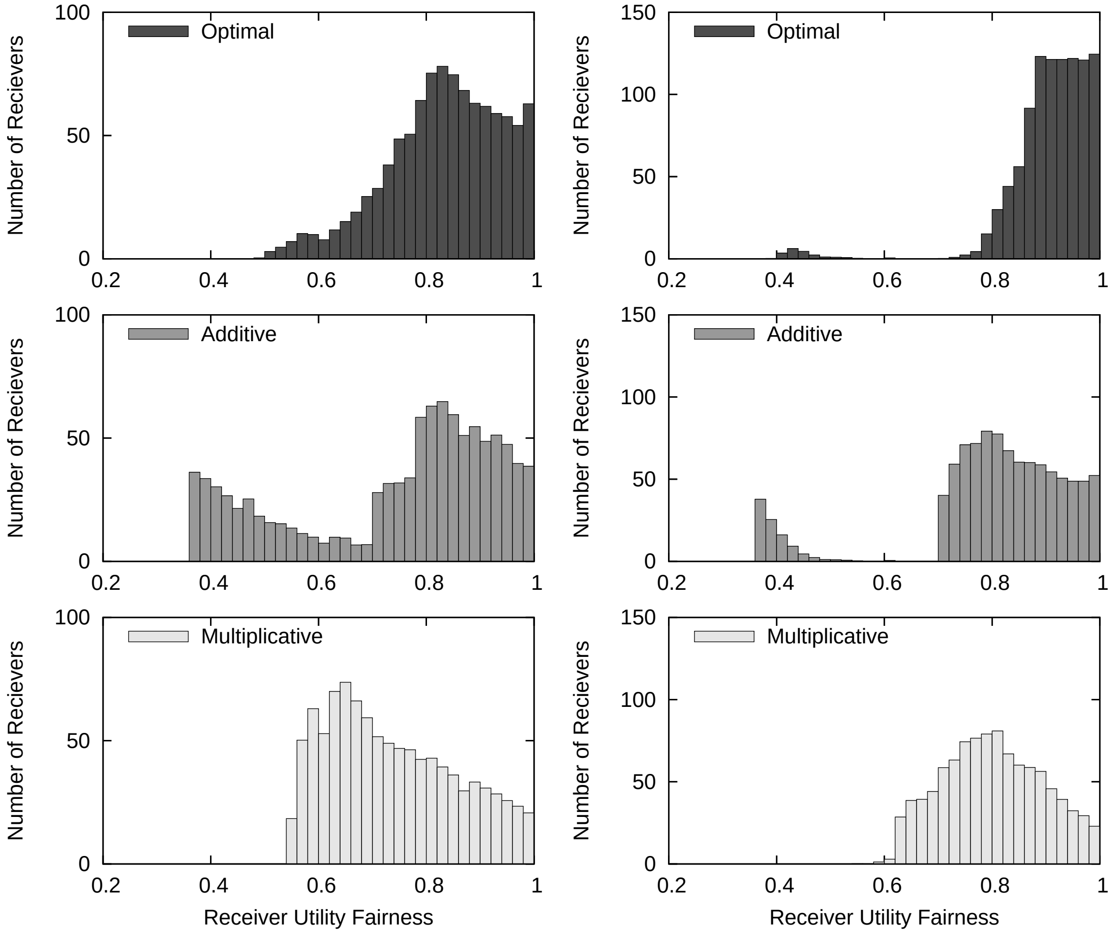
(c) multimodal distribution (d) measurement-based distribution

Figure 3.6: Performance in terms of MSE-based intra-session utility fairness achieved for different rate allocation strategies and distributions of receiver bandwidth capabilities.

ever the difference is approximately 29 % (see Figure 3.6(d)).

The above results prove the superiority of an optimal allocation scheme when compared to adaptive and multiplicative strategies. However, depending on the distribution of the receiver capabilities the performance difference by means of average receiver utility fairness might not always justify the computational overhead of the optimization algorithm. Thus, in Figure 3.7 and 3.8 we illustrate the distribution of the PSNR-based receiver utility fairness indices for the different distribution.

For all four distributions we observe that the optimal allocation scheme produces a lower degree of variations regarding the distribution of the fairness indices, which are clustered between 0.8 and 1.0. Thus, even in the case of measurement-based capability distributions the fraction of receivers with a very high degree of

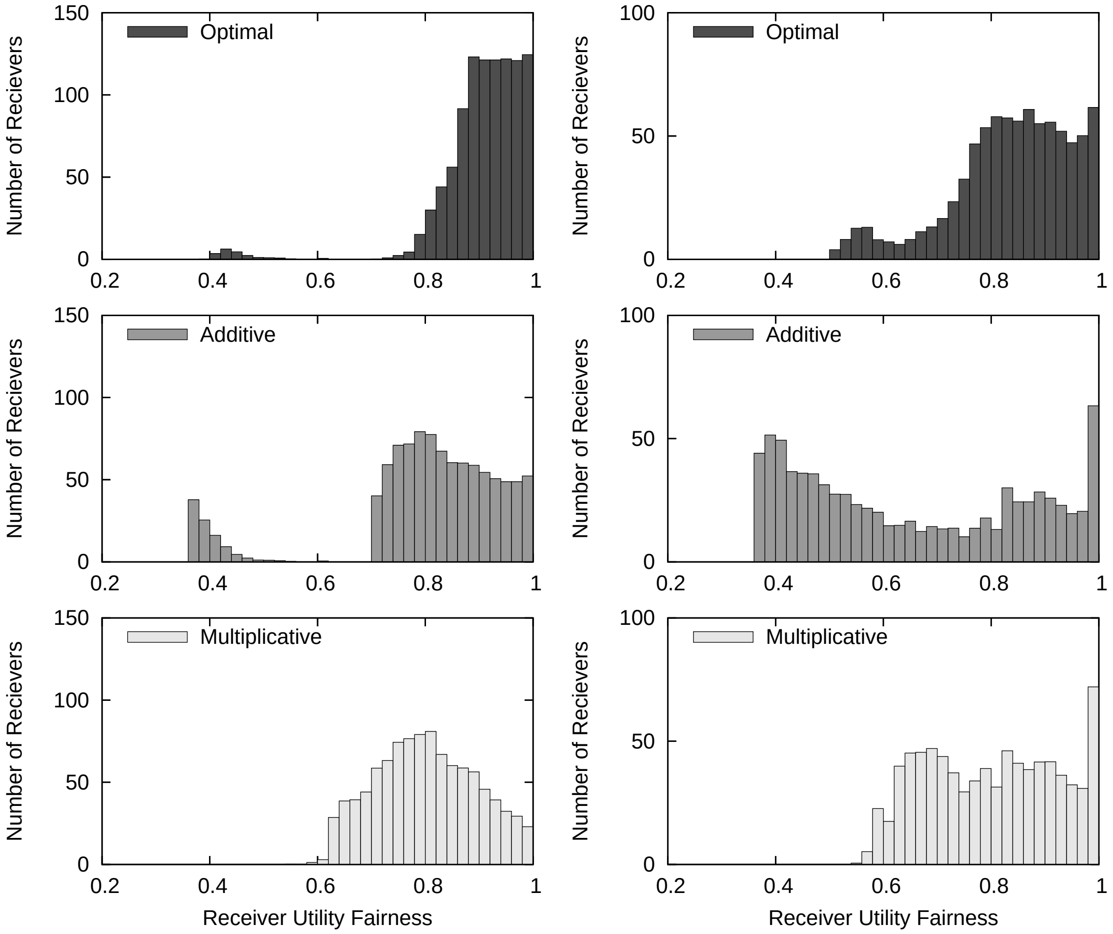


(a) uniform distribution (b) normal distribution

Figure 3.7: Distribution of the receiver utility fairness indices for different rate allocation strategies and distributions of receiver bandwidth capabilities if four multicast channels are allocated to the session.

experienced receiver utility fairness is larger than for both other distributions. For example, our algorithm calculates the four transmission rates such that approximately 65 % of the fairness indices are within the interval $[0.8, 1.0]$, while for the multiplicative strategy less than 50 % are within the same range.

Summarizing our results, we found that providing four multicast groups and optimizing their transmission rates seems to provide reasonable results. Our optimal rate allocation scheme generally outperforms static allocation strategies regarding the achieved average receiver utility fairness and the distribution of the individual fairness indices. The benefit compared to static allocation strategies depends on



(a) multimodal distribution (b) measurement-based distribution

Figure 3.8: Distribution of the receiver utility fairness indices for different rate allocation strategies and distributions of receiver bandwidth capabilities if four multicast channels are allocated to the session.

the underlying receiver capability distribution and ranges from 4 % to almost 40 %. Although for some distributions, the multiplicative strategy performs similarly regarding the average fairness, our strategy is much more flexible and adapts the transmission scheme to actual underlying conditions. This is an important feature in environments such as the Internet where network and receiver conditions are unpredictable and highly dynamic.

3.6 Summary

In this chapter, we have derived a model for capturing the fairness of a multicast receiver based on a general class of utility functions. While existing work formulates fairness in terms of network utilization, our model allows for capturing an application-aware measure. Based on our fairness definition, a performance metric for multi-rate multicast sessions has been devised and an algorithm for optimal allocation of the group transmission rates has been developed.

Subsequently, we adopted a recently presented square-root model for FGS-encoded video to derive a PSNR-based and an MSE-based instantiation of the user utility of streaming video. Although the assumed correlation of the objective measures with subjective assessment might be arguable, we emphasize that the goal of this work has been the provision of a general model for capturing an application-aware receiver utility fairness rather than an accurate subjective video quality metric. Our fairness index allows to easily adopt a utility function, for example, based on objective measures that correlate highly with subjective perception. Thus, the two instantiations of a utility function are intended to provide means for quantitative evaluation by approximating the expected range of accurate utility models.

Equipped with the PSNR-based and MSE-based utility functions, we performed a quantitative evaluation of our optimization algorithm. By means of simulations we showed that for scalable transmission rates from 128 kbps up to $2,560$ kbps a significant improvement of 25–40 % can be achieved when allocating four groups instead of a single multicast group. Furthermore, a comparison of the optimal allocation strategy with static strategies revealed that for heterogeneous receiver capabilities our scheme can significantly improve the intra-session performance for different distributions of the capabilities, which is not possible with static strategies.

Chapter 4

Feedback Control

The major focus of this chapter is on the development of a feedback mechanism that allows for collecting status information from receivers in a scalable way. The proposed feedback scheme is light weight and based on probabilistic sampling, thus, allowing the amount of feedback to be configured to a target level within some statistical bounds. More specifically, the server polls the members of a multicast session for feedback and each receiver replies with the probability signaled by the source. We therefore adopt an exiting technique for multicast session size estimation and develop a mechanism for dynamic adjustment of the estimated session size and feedback probability.

Using the proposed feedback scheme, the streaming server samples status information from the set of participating receivers. The server can then adjust the group transmission rates based on the sampled values in order to optimize intra-session fairness as discussed in Chapter 3. In an attempt to reduce the error introduced by data sampling, we develop a statistical model to approximate the receiver capability distribution based on the collected samples. For that purpose, we exploit existing work on Internet traffic measurement.

In the subsequent experimental analysis, the results of model-based optimization are evaluated and compared to the results achieved with an alternative approach. The latter relies on directly using the sampled report values to calculate the transmission rate vector. Particularly, we explore the influence of the sample size on the achieved intra-session fairness for several theoretical and measurement-based distributions of receiver bandwidth capabilities.

Figure 4.1 illustrates the roadmap of this chapter.

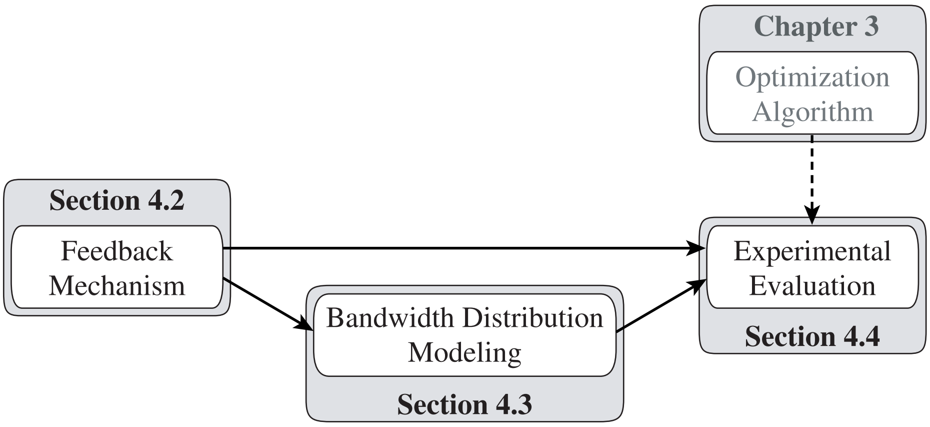


Figure 4.1: Roadmap of the chapter on *Scalable Feedback Control*.

4.1 Motivation

In Chapter 3 we developed a mechanism for optimally allocating and adapting transmission rates of a multi-rate multicast session. Thereby, sender-side knowledge of the receiver capability distribution has been assumed, which is required by the underlying optimization algorithm.

A scalable solution to estimate a receiver's actual status is only possible to be implemented at the receiver itself. Consequently, each receiver is assumed to have information about its bandwidth capability, for example, measured and calculated utilizing implicit network signaling (an appropriate technique will be introduced in Chapter 5). However, the streaming server needs to collect status information from the receivers by means of feedback in order to gather global knowledge. Once knowledge about the receiver capability distribution is available at the server, the latter can optimize the transmission rates of the session.

From the perspective of stream optimization, the sender would ideally solicit each receiver to send its actual status information. Though, requiring each receiver to frequently send report messages can easily lead to scaling problems. In the worst case, a multicast control scheme faces the well-known feedback implosion problem illustrated in Figure 4.2. First described by Danzig [Dan89], feedback implosion occurs when an entire multicast group synchronously or within a short term generates receiver reports that are sent back to the source. It can swamp nodes and links on the feedback path as well as the source, hence causing unstable conditions. Therefore, an efficient feedback control mechanism is necessary that keeps the feedback traffic bounded.

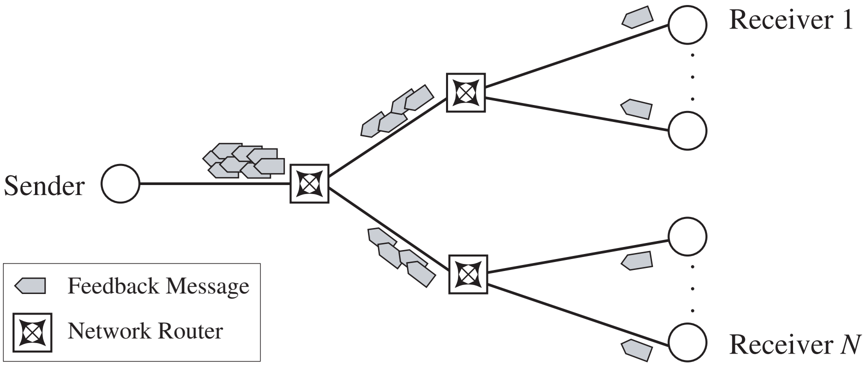


Figure 4.2: Feedback implosion resulting from a synchronously generated bunch of feedback messages.

4.2 Feedback Control Scheme

In order to bound the feedback traffic and avoid feedback implosion, the reporting time interval of each receiver might be eventually scaled down as proposed in the Real-Time Transmission Control Protocol (RTCP) [SCFJ03]. However, in that case the duration of a feedback round, that is, the overall time needed to collect an actual report of each receiver, scales linearly with the session size. Hence, the frequency of performing source-based stream adaptation decreases significantly for large groups. Furthermore, the fraction of status reports that becomes obsolete before a report round ends increases due to the dynamic nature of the Internet. This implies a higher imprecision with respect to the derived capability distribution and consequently a suboptimal stream organization.

Recall that the rate allocation algorithm in Chapter 3 depends only on the receiver bandwidth capability distribution. Since status reports of individual receivers are independent, using the common approach of timer-based suppression (see Section 2.6.3) is not very efficient. It requires each receiver to start a random timer before sending a feedback message and listen for the reports of other receivers on a multicast channel. If the observed feedback messages fulfill a certain condition before the timer expires (for example, the number of messages exceeds a given threshold), the corresponding receiver refrains from sending its status report.

With the aforementioned timer-based suppression approach, feedback decisions are subject to global knowledge. Thus, distribution of individual receiver reports to all participating hosts is required implying additional overhead. In contrast, our scheme relies on probabilistic sampling whereby individual feedback decisions are made independently. Furthermore, control of feedback bandwidth and sample size is provided within statistical bounds. A detailed description of the underlying techniques is given following overview of the design.

4.2.1 Design Overview

Our feedback scheme is based on the usage of a probabilistic approach for choosing a target number of receivers that are supposed to send status reports. Figure 4.3 summarizes the basic idea underlying the feedback control scheme by means of a flowchart.

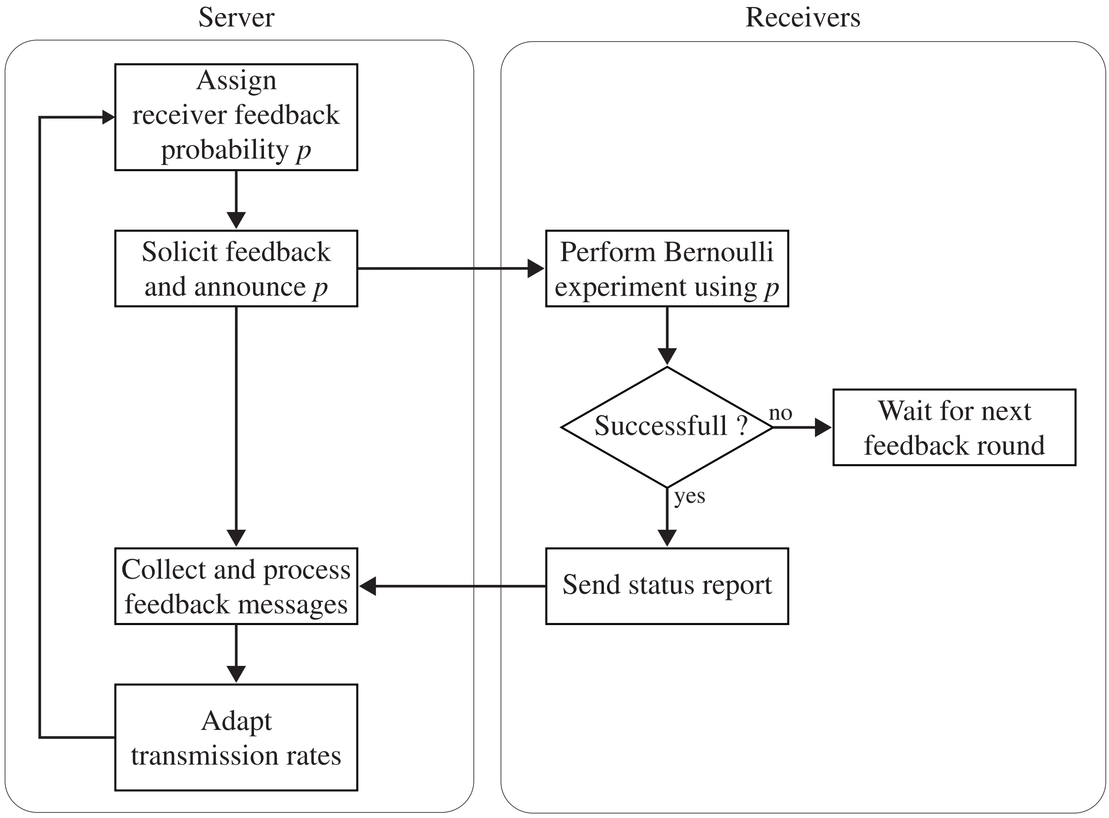

Server
Receivers
Assign receiver feedback probability p
Solicit feedback and announce p
Collect and process feedback messages
Adapt transmission rates
Perform Bernoulli experiment using p
Successfull ?
no
yes
Wait for next feedback round
Send status report

Figure 4.3: Flowchart overview of the feedback control scheme.

The selection of each receiver is performed independently in a distributed manner. Therefore, the sender announces a particular receiver report probability p to all participating end hosts. The announcement message triggers each receiver to perform a Bernoulli trial using the provided probability. If the outcome of a receiver's random trial is positive, it sends a status report reflecting its actual bandwidth conditions (i.e., its fair share) within a given feedback time interval. Otherwise, it refrains from reporting and waits for the next feedback round. Upon having collected the receiver reports, the server uses the sample for input to the optimization algorithm and adapts the transmission rates accordingly. The process is repeated periodically or according to any other strategy.

The main control parameter of our approach is the probability p for the success of a Bernoulli trial. For proper choice of value for the latter the target sample size and the actual session size are required. While the target report number is predetermined, the actual receiver population size is unknown and varies. It needs to be dynamically estimated at the server during the session for calculation of a reasonable feedback probability.

For the purpose of initial session size estimation, we adopted an end-to-end algorithm proposed Friedman and Towsley [FT99] that extends the work of Bolot et al. [BTW94]. The algorithm needs several polling rounds using a low and constant value for the feedback probability in order to calculate an estimate. This requirement makes the algorithm unfeasible to be applied frequently. Consequently, we develop an efficient algorithm for dynamically adjusting the session size estimate and the feedback probability in every feedback round. As a result, the feedback control process at the server consists of basically two parts:

1. **Initialization.** When the session size is unknown (or the estimate considered invalid/obsolete), feedback probability is set to a very low value. It is then increased after each feedback round until the number of feedback messages becomes sufficiently large. Having the sample size reached a threshold value, the algorithm aggregates feedback samples over several round at constant feedback probability in order to estimate the session size.

2. **Dynamic adjustment.** Using the estimated session size, the server probabilistically polls the receiver set for feedback. The actual sample size is then used to update the session size and the feedback probability in every feedback round. However, if a certain critical condition is violated indicating an invalid session size estimation, the feedback probability is reduced and the mechanism reenters the initialization step.

The sender-side control process is depicted Figure 4.4, which guides through the following discussion. However, before we continue with a detailed description

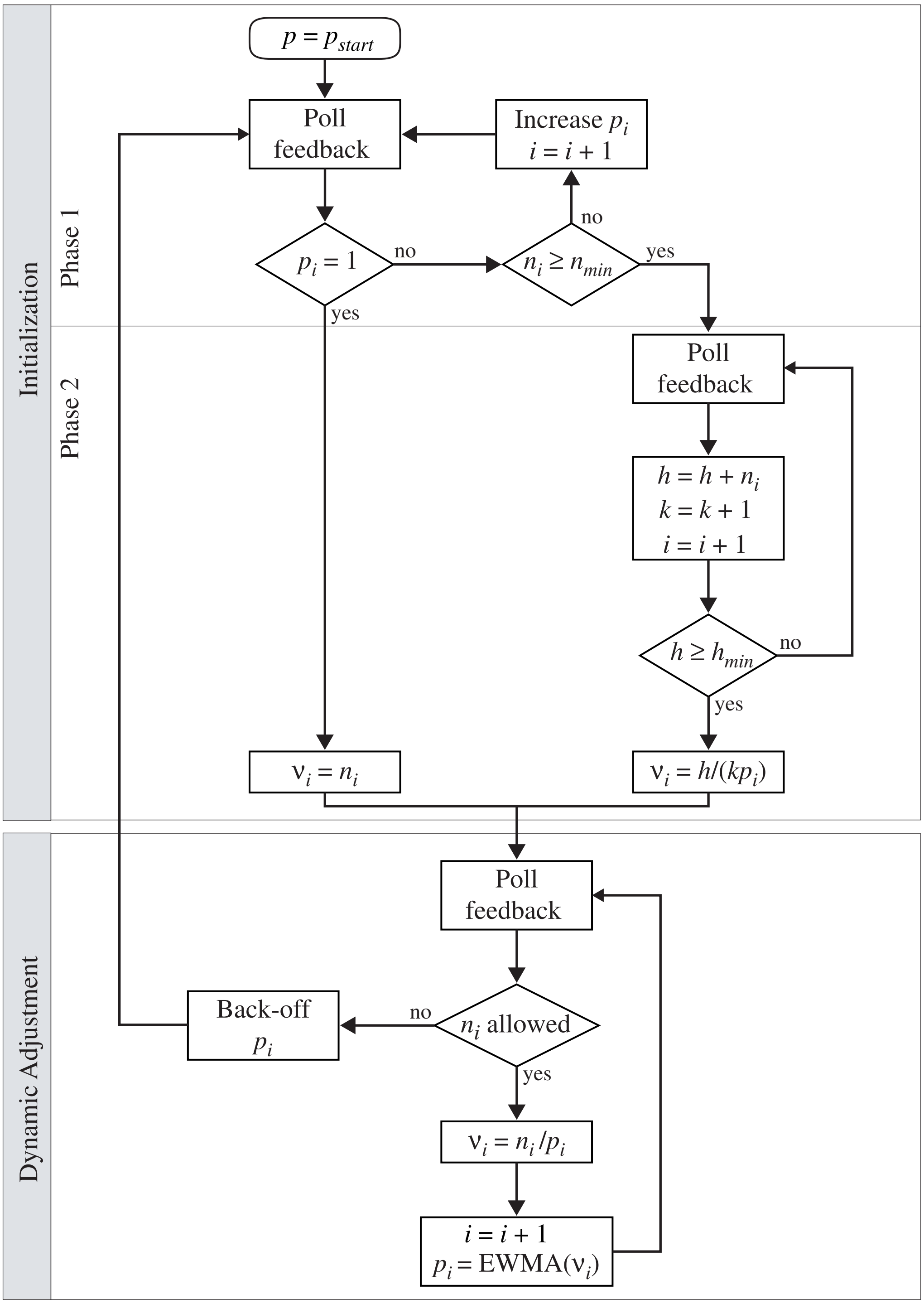


Figure 4.4: Flowchart illustration of the different phases of the feedback scheme.

of the control mechanism, we first discuss the underlying sampling process and its limitations in terms of precision.

4.2.2 Control Precision

The sampling process in each feedback round is mathematically described as an experiment of independent Bernoulli trials. Each receiver trial is modeled by a random variable, which can take only two values: 1 ("success") for sending feedback, and 0 ("failure") for refraining from reporting. In the remainder, we refer to the probability of success as the success probability p. Since all receivers perform identical but independent trials, the number of feedback messages sent per round is modeled by a random variable X that follows a binomial distribution $BD(N;p)$. The corresponding probability function is given by:

$$\Pr(X = n) = \binom{N}{n} p^n (1-p)^{N-n}, \tag{4.1}$$

where N denotes the number of receivers performing the experiment (i.e., the population size), and p the success probability. The expected value and the variance of X are computed as follows:

$$\begin{aligned} E(X) &= Np, & (4.2)\\ V(X) &= Np(1-p). & (4.3) \end{aligned}$$

Considering a target number of feedback messages n^* and knowledge about the receiver population size N, the success probability can then be calculated using Equation 4.2 as $p = \frac{E(X)}{N} = \frac{n^*}{N}$.

Normally, a multicast session is expected to have a reasonable large population size. Otherwise, the multicast transmission mode is not very efficient and scalability is not an issue. Furthermore, a valid assumption is to consider the target number of feedback messages n^* reasonable high so that $V(X) > 9$ holds. The latter is a rule of thumb [Sac02] to apply the well-known De Moivre-Laplace theorem. It states that a normal distribution provides a very close approximation to the binomial distribution when N is large and Np is not extremely close to either 0 or N. Thus, for further analysis we can approximate the distribution of the number of feedback messages by a normal distribution $N(\mu;\sigma)$ with a probability function defined as follows:

$$\Pr(X = n) = \frac{1}{\sigma_X\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{n-\mu_X}{\sigma_X}\right)^2}, \tag{4.4}$$

where $\mu_X = E(X) = Np$ and $\sigma_X^2 = V(X) = Np(1-p)$.

The above form allows to easily calculate the statistical bounds for the expected sample size. More precisely, the two-sided confidence interval (CI) for which there is a probability $1-\alpha$ that the number of reporting receivers is within the limits of the interval follows:

$$\Pr(|X - n| \leq \varepsilon) = 1 - \alpha, \tag{4.5}$$

where

$$\varepsilon = z_{1-\alpha/2}\sqrt{Np(1-p)}, \tag{4.6}$$

is the interval's half-length and $z_{1-\alpha/2}$ is the $(1-\alpha/2)$ quantile of a unit normal variate. Figure 4.5 shows that for a significance level of $\alpha = 0.01$ the interval's half-length is less than 30 over a broad range of the population size and reasonable target sample sizes[1].

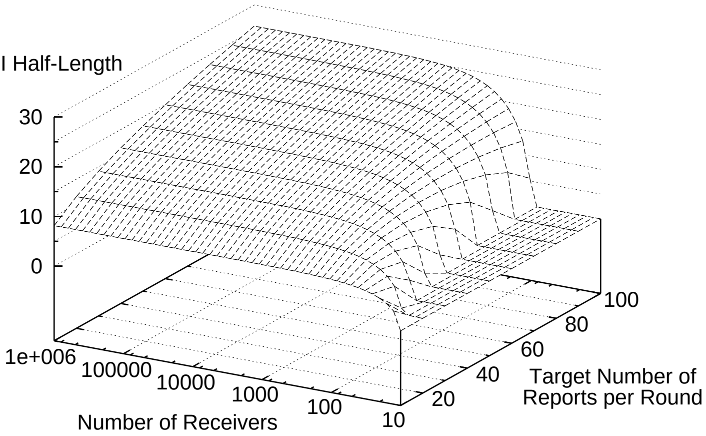


Figure 4.5: Half-length of the double-sided confidence interval for a significance level of $\alpha = 0.05$.

Providing a numerical example, with a target sample size of $n^* = 50$ in 99 % of the feedback rounds the actual sample size is expected to be within the interval $[32, 68]$. The results indicate that the probabilistic sampling approach provides sufficient statistical bounds for the purpose of feedback suppression in the context

---

[1] The study of a reasonable sample size follows in Section 4.4.

of this work.

4.2.3 Initialization

For a proper choice of the feedback success probability, knowledge about the session size is required at the server. A corresponding mechanism for that purpose is provided by RTCP [SCFJ03] when the latter is used as the underlying control protocol. The mechanism requires feedback from all participants increasing the feedback time interval for scalability reasons as the session size increases. Hence, it implies a high delay and the quality of the estimate becomes rather poor for large groups.

For initial session size estimation in our feedback control scheme, we adopt the algorithm originally proposed by Bolot et al. [BTW94] and extended by Friedman and Towsley [FT99]. It basically consists of a two-phase iterative process as depicted in Figure 4.4. The corresponding variables and parameters used are:

i – index of the polling/feedback round,
p_i – feedback probability in the ith round,
n_i – number of feedback messages (sample size) in the ith round,
n_{min} – threshold value for the sample size for the transition to phase 2,
k – count of successive polling rounds in phase 2,
h – aggregate number of feedback messages over all successive feedback rounds in phase 2,
h_{min} – the minimum number of feedback messages required for estimating the session size with a certain quality,
ν_i – estimate of the session size in round i.

Phase 1

At the beginning of phase 1 the source has no valid estimate of the session size. Hence, in the first round it starts polling with a conservative value for p that is expected to hardly cause any receiver to send feedback. Subsequently, p is increased in each polling round until $p = 1$ or the actual feedback sample size n reaches the given threshold n_{min}. In the latter case, the mechanism transits to phase 2; in the former case, the estimated session size ν is set to the actual sample size and phase 2 is skipped. An efficient algorithm for adapting p in order for n to approach n_{min} in only a few rounds has been proposed by Bolot et al. [BTW94] as follows:

$$p_i = \frac{2^{i-2}}{p^{-1} - 2^{i-2}}. \tag{4.7}$$

Phase 2

In the second phase, the feedback probability p is fixed to the last value that caused phase 1 to terminate and it is used in the subsequent feedback rounds for polling. In each of the following rounds of phase 2 counter k is incremented and the number of feedback messages received is added to the variable h. This process is repeated until h reaches a predetermined threshold value h_{min}.

Since p is kept constant over all feedback rounds of phase 2, the underlying process is an experiment of k independent and identical distributed trials. Thus, the mean value for the sample size in phase 2 can be calculated as $\bar{n} = \frac{h}{k}$ and assumed as coming from a normal distribution. Using Equation 4.2 the session size can then be estimated as:

$$\nu = \frac{\bar{n}}{k} = \frac{h}{kp}, \tag{4.8}$$

with the corresponding half-length of the confidence interval given by[2]:

$$\varepsilon = \frac{z_{1-\alpha/2}}{2kp} \left((1-p)\, z_{1-\alpha/2} + \sqrt{4k\bar{n}\,(1-p) + (1-p)^2\, z^2_{1-\alpha/2}} \right). \tag{4.9}$$

For a reasonable dimensioning of h_{min}, we apply a measure that captures the quality of the estimator as the ratio of the interval's half-length ε and the magnitude ν of the point estimator:

$$\varphi = \frac{\varepsilon}{\nu}. \tag{4.10}$$

Substituting ν and ε with Equation 4.8 and Equation 4.9, respectively, Equation 4.10 can be solved for h. Given a target estimation quality φ^* and assuming a large population size, Friedman and Towsley proved that h_{min} can be approximated by:

$$h_{min} \approx \frac{(1+\varphi^*)\, z^2_{1-\alpha/2}}{(\varphi^*)^2}. \tag{4.11}$$

To conclude with an example, for a confidence level of 95 % the z-value of the unit normal distribution is $z = 1.96$. To achieve an estimation quality of $\varphi^* = 0.1$

---

[2] A derivation of the interval's half-length can be found in Friedman and Towsley [FT99].

it follows from the above that $h_{min} \approx 423$. That is, in this example at least 423 feedback messages need to be received in phase 2 before the session size can be estimated with the required precision.

4.2.4 Dynamic Adjustment

Upon having a reliable estimate of the session size, the server can calculate an appropriate success probability for the Bernoulli experiments and provide it to the receivers in the next feedback round. However, the session size of a multicast session is subject to dynamic changes since receivers may join and leave the session. Consequently, the algorithm described in the previous section would have to be repeated over and over again. The disadvantage of such an approach is obvious, since both phase 1 and phase 2 normally will require several feedback rounds before an estimate can be calculated.

To overcome the aforementioned shortcomings we develop an algorithm that dynamically adjusts the session size estimate and the success probability in each feedback round (see Figure 4.4). However, since the underlying polling process is random, relying only on the data of the most recent feedback round for adjusting p might easily lead to oscillations. Therefore, we propose the use of an estimator based on the Exponentially Weighted Moving Average (EWMA) of the session size estimate in order to stabilize the process:

$$\nu_i = \kappa \frac{n_{i-1}}{p_{i-1}} + (1-\kappa)\,\nu_{i-1}, \tag{4.12}$$

where κ denotes the EWMA weight parameter. The initial values for ν, n, and p are set to the mean values obtained from phase 2.

With the usage of the EWMA estimator, the feedback schemes adjusts the feedback probability for the next feedback round by dividing the target feedback sample size by the smoothed session size estimate. The process is repeated for each feedback round. However, the mechanism can be configured in order to terminate in case of a critical condition. The latter might apply when the underlying group dynamics produce sample sizes larger or lower than a configured maximum or minimum value, respectively.

Parameter Setting

The crucial parameter in the presented approach is κ, which when being configured requires to consider the conflicting requirements of robustness and responsiveness:

- A low value of κ increases the robustness to random variations since new samples are less weighted. Thus, in steady state setting κ to a low value is preferred.
- A high value of κ increases its sensitivity to sudden changes in population size. Thus, for very dynamic groups setting κ to low values close to 0 is preferred.

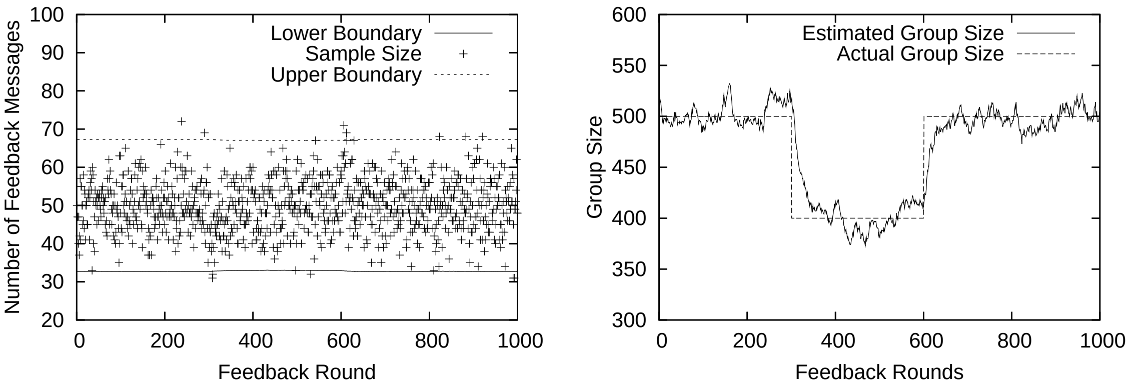
Finding an optimal value for the above problem analytically is not feasible. It requires knowledge about group dynamics, for which empirical data is lacking to our knowledge. Thus, we briefly discuss the parameter setting by the results obtained from our experiments.

The experimental setup consist of sets of the results of $1,000$ consecutive feedback polls, which have been generated for a small session size of $N_1 = 500$ receivers and a large session size of $N_2 = 10,000$ receivers. The target feedback sample size for both scenarios is set to $n^* = 50$, which is a reasonable size as we discuss in Section 4.4. Considering the aforementioned requirements of robustness and responsiveness, the session size in the experiments exhibits sudden and pronounced changes: after round 300 the group decreases by $20\,\%$, and switches to the initial session size again after round 600.

For the described configuration setup, heuristically setting $k = 0.1$ provides a reasonable trade-off between smoothness and responsiveness. Figure 4.6 and Figure 4.7 show the results for both configurations. We observe that in both cases the algorithm is reactive so that the fluctuations in sample size caused by the disturbance at feedback round 301 and 601 are not significantly increased. Moreover, the error in session size estimation takes a negligible level.

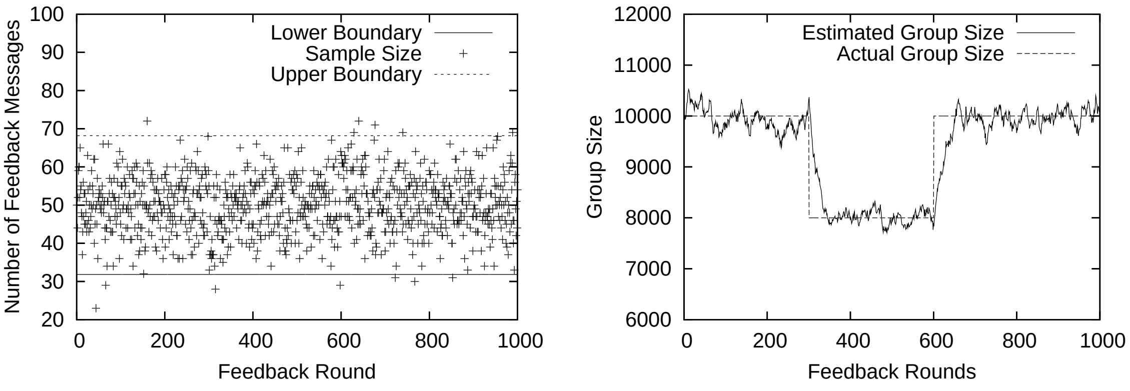
4.3 Approximation of Bandwidth Distributions

Recall that feedback polling in the context of this work is required for the adaptation of transmission rate allocation as described in Chapter 3. Therefore, the server needs a representative sample of the bandwidth capabilities of the receiver



(a) sample size (b) session size

Figure 4.6: Dynamic behavior of the session size estimator for a population size $N = 500$ receivers and a target sample size $n^* = 50$.



(a) sample size (b) session size

Figure 4.7: Dynamic behavior of the session size estimator for a population of $N = 10,000$ receivers and a target sample size $n^* = 50$.

population. This is guaranteed by the proposed feedback mechanisms since every receiver performs an identical and independent Bernoulli trial.

The aforementioned characteristics of the probabilistic feedback approach allows the server to perform optimization of the transmission rates based on the representative feedback sample. We refer to this approach in the following as *pure sampling.* However, since each sample stems from a random process, it will introduce a certain random error to the outcome of the optimization. The error can be reduced and the accuracy of the transmission rate optimization increased by choosing a larger sample size. However, this implies additional feedback overhead.

An alternative approach that we consider is the modeling of the empirical data by statistical distributions. If the server has available an appropriate model that resembles the bandwidth distribution of the receiver population, it might estimate the model parameters utilizing a representative sample. Optimization of the transmission rates can then be performed based on the sample. The closer the model comes to the empirical distribution, the higher the accuracy of the optimization.

However, in order to find an appropriate model for the bandwidth distribution, empirical data about Internet traffic in terms of available bandwidth needs to be collected and analyzed. Moreover, the distribution of bandwidth capabilities within multicast sessions needs to be considered. Since wide-area deployment of multicast services is still to emerge, empirical studies are lacking to our knowledge. Hence, precise modeling of the bandwidth capability distribution is a very challenging task and out of the scope of this thesis.

The objective of this section is to study a simple yet flexible model that can be exploited for the aforementioned purpose of rate optimization. Therefore, we resort to measurement studies conducted by other Internet researchers. Particularly, we selected the results presented by Paxson [Pax97b] and data from the PingER (Ping End-to-end Reporting) project [MC00] (see Appendix B.5). Based on the empirical distributions a simple statistical approximation is studied.

4.3.1 Methodology

In order to find a simple yet appropriate statistical model for approximating the empirical data, the characteristics of the latter have to be considered. Interestingly, although the studies of Paxson and the PingER project have been performed with a relatively large time lag and over different environments, both lead to similar results regarding the shape of the data distributions. Nevertheless, they differ in terms of the scale: compared to the Paxson study, the PingER data reflects advances in network technology such as increased link capacities since it has been obtained only recently. From the characteristics of the empirical distributions the following criteria for the model have been derived:

- The model should be skewed to the left since both empirical distributions are L-shaped.
- The model should be bounded to non-negative values since, obviously, band-

width is also non negative.

- The model should be flexible to adapt to empirical data with varying characteristics, such as scale, skewness, and amplitude of mode.

Consideration of the above requirements led to preselection of two statistical distributions for further investigation:

- **Gamma distribution.** The gamma distribution is defined by a scale parameter a and a shape parameter b. Its probability density function is defined as:

$$pdf(x) = \frac{\left(\frac{x}{a}\right)^{b-1} e^{-\frac{x}{a}}}{a\Gamma(b)}, \tag{4.13}$$

  where $\Gamma(\cdot)$ is the gamma function.

- **Weibull distribution.** The Weibull distribution is also defined by a scale parameter a and a shape parameter b with the probability function

$$pdf(x) = \frac{bx^{b-1}}{a^b} e^{-\left(\frac{x}{a}\right)^b}.$$

These statistical models are very flexible and can be parameterized to model other distributions such as the beta distribution, exponential distribution, Erlang distribution, and normal distribution. The procedure used for evaluating both distributions for the approximation of the empirical data consist of three steps:

1. Samples from the empirical distribution are generated from the Paxson results and the PingER data. In order to study the influence of the sample size on the accuracy achieved in modeling the empirical distribution, several sample sizes have been chosen: 30, 50, 100, 500, and $1,000$.

2. Assuming the samples as coming from a gamma and a Weibull distribution, parameters of the corresponding model are estimated using Maximum-Likelihood-Estimators (MLEs) for each sample size.

3. Finally, Kolmogorov-Smirnov (KS) tests are conducted in order to quantify the accuracy with which the parameterized model approximates the underlying empirical distribution of the sample.

Note that the final results of our study are based on 15 repetitions of the procedure, which we conducted in order to reduce the impact of random effects.

Maximum-Likelihood-Estimation

In the second step of the procedure we use the Maximum Likelihood method to derive point estimators for the unknown parameters of a statistical distribution. We briefly summarize the basic theory that underlies the well-known MLE method and refer for further information to literature on mathematics (e.g., [Pap01]).

If X denotes a random variable whose probability density function contains the yet unknown parameter ψ, the likelihood function is given by:

$$L = L(\psi) = pdf(x_1; \psi) \cdot pdf(x_2; \psi) \cdots pdf(x_n; \psi), \tag{4.14}$$

where n denotes the sample size and *pdf* the probability density. The estimate $\hat{\psi}$ for the unknown parameter ψ is the value that maximizes the Likelihood function $L(\psi)$. Since the *pdf* of the gamma and Weibull distribution are continuous functions, $\hat{\psi}$ can be derived by applying the condition $\frac{d}{d\psi}L = 0$ for the maximum and rearranging the result.

Upon having estimated the parameters of the statistical distribution for approximating the empirical data, in the next step a KS goodness-of-fit test is performed. The objective is to test the goodness-of-fit of the model and the empirical distribution of the sample it is derived from.

Kolmogorov-Smirnov Test

The KS test is a very common technique for testing the goodness-of-fit for small sizes and ungrouped data. It requires the assumed statistical distribution functions to test for to be continuous, which is given for the gamma and the Weibull distribution. The procedure for each sample is then as follows:

1. The empirical cumulative distribution function (ecdf) is created from the sample values.
2. The parameters obtained by means of the MSE method are used to create the cumulative density function (cdf) of the corresponding gamma and Weibull distribution, respectively.
3. The largest occurring distance d between the cdf and the ecdf is calculated as the test statistic for the hypothesis that the sample data follows the tested distribution:

$$d = \max_{1<i<n} \left\{ \left| cdf_i - \frac{i}{n} \right| \right\}. \tag{4.15}$$

Sample Size	Critical Value	Mean Distance		Rejections	
		Paxson	PingER	Paxson	PingER
30	0.218	0.14	0.18	0	3
50	0.172	0.12	0.16	1	6
100	0.122	0.10	0.15	2	13
500	0.055	0.09	0.14	15	15
1,000	0.039	0.08	0.14	15	15

Table 4.1: KS test results for the gamma distribution.

Sample Size	Critical Value	Mean Distance		Rejections	
		Paxson	PingER	Paxson	PingER
30	0.218	0.13	0.18	0	2
50	0.172	0.12	0.16	0	5
100	0.122	0.10	0.15	1	13
500	0.055	0.08	0.14	15	15
1,000	0.039	0.08	0.14	15	15

Table 4.2: KS test results for the Weibull distribution.

4. For a given significance level α the critical value for the distance is determined from KS-test tables. If the test statistic exceeds the critical value, the hypothesis regarding the distribution function is rejected.

4.3.2 Results

The results of the model-based approximation of the empirical distributions are presented in Table 4.1 and Table 4.2 for the gamma and the Weibull distribution respectively. They show the test statistic (distance) averaged over the 15 trials of the independent trials and the number of trials of each experiment for which the hypothesis had to be rejected. The critical values for a significance level of $\alpha = 0.1$ are obtained from literature.

From the obtained experimental results, we can conclude that the Weibull approximation fits slightly better than the gamma model for both the Paxson as well as the PingER samples. In the case of smaller sample sizes the hypothesis is never rejected for the Paxson data. A sample size of 100 led to rejection of the hypothesis only once out of 15 trials. In contrast, for the PingER data the rejection rate is significantly higher and increases suddenly for a sample size between 50 and 100. The results clearly show that for sample size of 500 and above the hypothesis has to be rejected for all samples stemming from both data sets. This is explained by the inherent characteristics of the underlying data sets the samples have been drawn from. The population of the available bandwidth values does not follow a well-defined statistical process. It is rather a mixture of random processes and certain constrains posed from the technological environment measurement (e.g., access technology).

Nevertheless, recall that the idea for studying a modeling approach in the context of this work is not the derivation of an exact model. The intention is to investigate whether the results of the sender-side rate adaptation process (see Chapter 3) can be improved when using a model-based approach based on sampling instead of the pure sampling approach. If so, it would allow for feedback reduction while keeping the optimization results at a similar level.

In the following experimental evaluation we use a Weibull-based approximation as an alternative approach to pure sampling. The experiments should indicate whether and when it is reasonable to introduce an additional component in the feedback scheme that derives a model-based approximation of the bandwidth capability population from the status reports of a single feedback sample.

4.4 Experimental Evaluation

The incentive for using feedback in this work is to provide the server with receiver status reports that it can use for the purpose of rate adaptation. Therefore, we have developed a feedback mechanism as described in Section 4.2 that allows for regulating the feedback volume within statistical bounds by means of probabilistic sampling. In the focus of this section is an experimental study on how the sample size influence the results of the sender-side rate adaptation process using the optimization algorithm from Chapter 3.

4.4.1 Experimental Setup

The procedure for the experimental study is as follows:

1. Populations of $N = 10,000$ receiver bandwidth capabilities are generated to resemble the case of a large multicast. Basically, six populations are generated that come from different distributions as described in Appendix B: uniform,

normal, multi-modal, Weibull, measurement-based Pasxon, and measurement-based PingER. For each population we calculate the optimal transmission rates and the resulting intra-session utility fairness U^* (see Chapter 3).

2. Samples of target size $n^* \in \{30, 50, 100, 500, 1000\}$ are drawn from each population according to the statistical process previously described. This simulates the feedback control mechanism when each receiver reply to a feedback request with a probability $p = \frac{n^*}{N}$.

3. Subsequently, our optimization algorithm from Chapter 3 is applied to optimally allocate the transmission rates. Thereby, we apply two alternative strategies:
    (a) We use the sample values directly, which we refer to as *pure sampling.*
    (b) We assume the underlying distribution can be described by a Weibull function reasonably well. Thus, we calculate the Weibull parameters using the MLE approach (see Section 4.3) and generate $10,000$ values with this setting, which are then used instead of the original sample. We refer to this approach as *statistical modeling.*

4. Finally, the achieved intra-session utility fairness is calculated and compared to the optimal value U^* obtained with global knowledge.

Each experiment has been repeated 20 times in order to draw conclusions about the statistical significance of our results. In addition to the mean value for the intra-session utility fairness $\bar{U}$ over 20 repetitions, we also use:

- One-sided confidence interval for a significance level $\alpha = 0.01$. Since feedback samples from each repetition comes from the same population, the mean over all repetitions follows a student distribution (also known as t-distribution). The confidence interval is calculated to give the lower bound for the expected mean $\bar{U}$ for a confidence of 99 %.

- One-side tolerance interval for a significance level $\alpha = 0.05$. The intra-session utility fairness U of a single sample is a measure depending on all values of the samples. Thus, it is not normally distributed and to draw conclusions about the expected range of U we resort to the concept of tolerance intervals. It gives the interval that with 95 % confidence will cover 86 % of all U of a sample.

4.4.2 Discussion of Results

Figure 4.8 summarizes the results of the conducted experiments. Note that the mean values as well as confidence and tolerance metrics are presented each in terms of the normalized difference (error) to the optimal value U^*.

For the uniformly distributed data sets and small sample sizes $n = 30$ and $n = 50$ applying the Weibull approximation to the samples leads to better performance than pure sampling. Only as the sample size increases to very high values $n = 500$ and higher, the pure sampling provides better results (see Figure 4.8(a)).

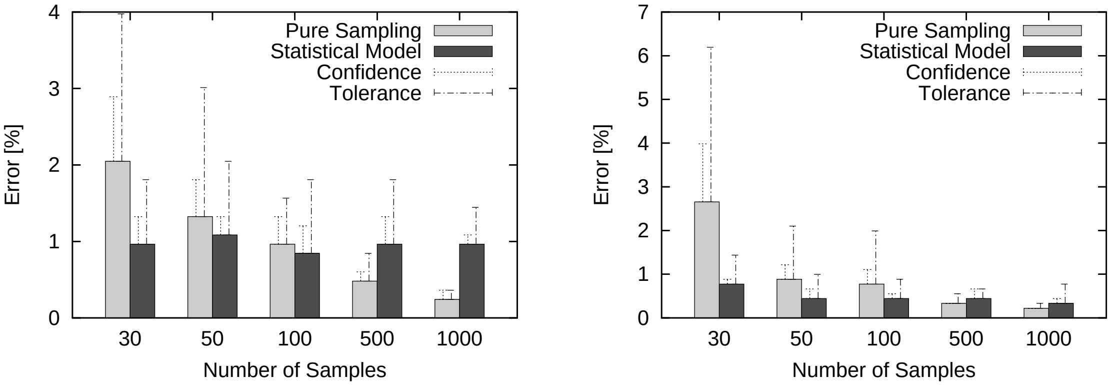
As Figure 4.8(b) shows, in the case of normally distributed receiver populations and samples of a size up to $n = 100$ the model-based approach clearly outperforms pure sampling. This especially holds for small sample sizes of $n = 30$ and $n = 50$. The mean error for the intra-session utility fairness and the tolerance of the model-based approach are below 1 % and 2 %, respectively, for all sample sizes. The supreme results are attribute to the flexibility of the Weibull distribution, which can be parameterized to closely approximate a normal distribution.

Approximation of samples from multi-modal distributed populations are very challenging and lead to better results using pure sampling (see Figure 4.8(c)). The Weibull distribution has only a single mode and fails to approximate the underlying multi-modal distribution sufficiently. While the mean error of the model-based approach for all other distributions is below 2 % here it reaches values of up to 5 %. For a reasonable small sample size of $n = 50$ the tolerance interval reaches even a value of 7 %. Thus, in the case that the population is clustered the pure sampling approach should be preferred.

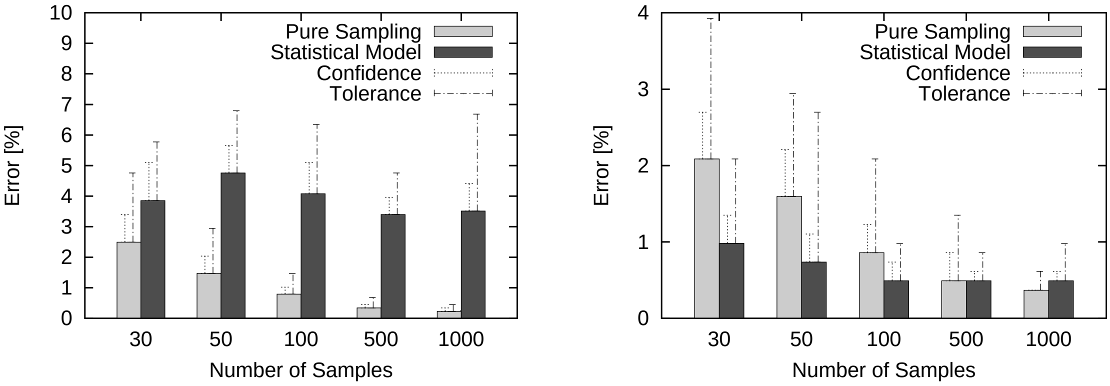
As expected, for Weibull distributed populations the model-based approach significantly outperforms the pure sampling approach for reasonable sample sizes (see Figure 4.8(d)). The error is bounded to very similar values as in the case of samples from normally and uniformly distributed populations. In contrast to the latter, even for large samples pure sampling cannot provide better results.

For the measurement-based populations the approximation provides only slightly better results for sample sizes $n = 30$ and $n = 50$ (see Figure 4.8(e) and Figure 4.8(e)). As the samples become larger, pure sampling shows better performance. This correlates very well with the KS test results from Section 4.3.2 where larger sample sizes lead to a high rejection rate.

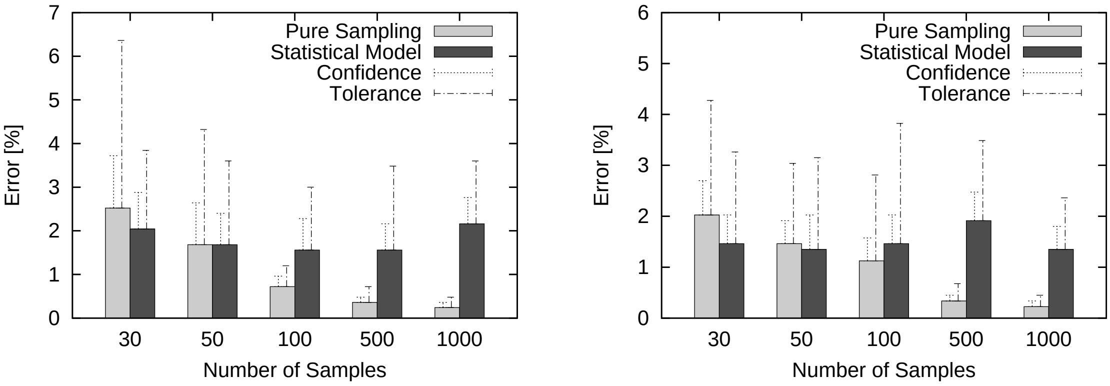
As a conclusion, the experimental results indicate that a reasonable target sample size for the feedback control scheme is in the region of $n^* = 50$. Assuming a



(a) uniform distribution (b) normal distribution



(c) multi-modal distribution (d) Weibull distribution



(e) Paxson measurement (f) PingER measurement

Figure 4.8: Experimental results for pure sampling and model-based rate adaptation for different feedback sample sizes and bandwidth distributions.

low feedback collection time of 1 s and feedback packets of 125 Bytes the resulting burst in feedback traffic only has a rate of 50 *kbps*.

4.5 Summary

In this chapter, we have developed a multicast feedback control scheme based on probabilistic sampling. Our scheme allows to regulate the feedback traffic to a target level within statistical bounds. For appropriately setting the feedback probability in each polling round, the actual session size is estimated with a two-step approach. In the first step, the session size estimate is initialized adopting a state-of-the-art mechanism. In the second phase, the mechanism is extended by a light-weight algorithm for updating the estimate in each feedback round. The algorithm is tunable and by means of experiments we proved its smoothness and responsiveness.

Subsequently, a model-based approach for deriving global information from the sample values has been investigated for enhancement of the sender-side rate adaptation process. Therefore, existing work of Internet bandwidth measurements has been studied and an approximation based on Weibull distributions has been proposed.

With simulations we systematically explored the influence of the sample size on the achieved intra-session fairness for several theoretical and measurement-based distributions of receiver bandwidth capabilities. Therefore, the intra-session fairness metric and the optimization algorithm from Chapter 3 have been utilized. We also compared the sender-side rate adaptation performance when feedback messages are directly used for optimization to the case where prior to optimization a Weibull-based approximation is carried out. The results indicate that the latter approach can improve the intra-session performance for small sample sizes, however, not in every case.

Chapter 5

Equation-Based Fair Share Estimation

In this chapter, we investigate appropriate techniques for estimating the fair share of a receiver participating in a multicast streaming session. We begin with the introduction of the underlying fairness criterion, namely TCP-compatibility. This criterion has been accepted as a de facto requirement for alternative Internet transport protocols in order to control flows that fairly allocate and share network resources with TCP traffic.

Since this thesis addresses streaming applications that require smooth transmission rates, it is based on the state-of-the-art equation-based approach. While aggressive probing techniques produce pronounced rate fluctuations, the equation-based approach builds on an analytical model of TCP to enable smoother behavior. It calculates an estimate of the steady-state throughput of a TCP-regulated flow based on loss and round-trip time estimations. The accuracy of rate estimation relies in particular on the employed loss estimation algorithm, for which existing techniques (recently proposed for standards track in the IETF [HFPW03]) are adopted.

By means of extensive network simulations we evaluate the applicability of the original algorithm. We identify and analyze its shortcomings, especially its dependency on the transmission rate. Our results indicate that naively adopting the original algorithm might not necessarily provide the expected behavior. This observation leads to the development of modifications that improve the rate estimator. Using network simulations, the performance improvement with respect to the precision of the estimator is shown and its suitability for our target scenario of multicast streaming is discussed.

Figure 5.1 depicts the roadmap of this chapter.

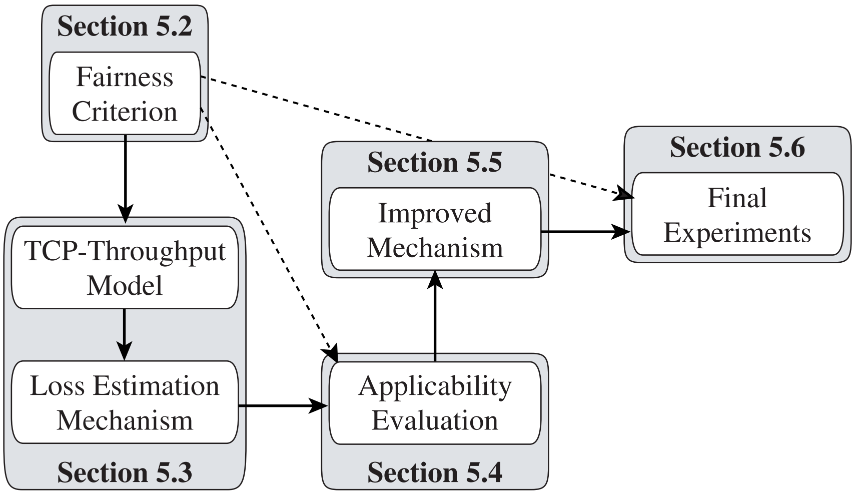


Figure 5.1: Roadmap of the chapter on *Equation-Based Fair Share Estimation*.

5.1 Motivation

Multi-rate multicast transmission schemes that follow Internet's end-to-end paradigm can involve both the sender and the receivers for the purpose of rate adaptation. In order to accommodate heterogeneous conditions the source can optimize the transmission rates, for example, using the algorithms and mechanisms devised in Chapter 3 and Chapter 4, respectively. On the other hand, group subscription management in the Internet is receiver-driven requiring each receiver to individually join and leave the appropriate multicast channels.

Hence, key components for both sender-side rate optimization and receiver-side rate control are techniques for inferring the fair bandwidth share of each receiver. Addressing this issue in a scalable way is only reasonable with distributed receiver algorithms, which build the core of a scalable multi-rate scheme.

For streaming applications, approaches based on analytically modeling the long-term TCP throughput are very well suited. The benefit of equation-based estimation of the fair share is two-fold: on one hand, multicast receivers can base their join and leave decisions on the calculated rate instead of having to frequently probe for the appropriate subscription level, as we discuss in Chapter 6; on the other hand, the source might adapt and optimize the transmission rates according to receivers' feedback, as presented in Chapter 3.

While equation-based fair share estimation is a promising approach originally designed in the context of closed-loop unicast control, its applicability to multicast schemes has not been studied sufficiently.

5.2 Fairness Criterion

The Internet provides only a best-effort service and network elements do not globally exert active control over resources. Due to the lack of network mechanisms that force fair sharing of the available resources, end systems have to employ appropriate control mechanisms. As a result, bandwidth allocation becomes a function of the control mechanisms employed by the competing end systems. The latter are expected to adopt the "social" rules implied by TCP and be cooperative by reacting to congestion indication and adapting their transmission rates properly and promptly. This paradigm of passive routers and active hosts has been very successful in the Internet, where TCP-based traffic dominates [FML+03]. TCP's congestion management mechanisms have been keys to the stability of the Internet, despite its rapid growth in traffic with respect to volume and diversity.

TCP provides congestion control mechanisms and serves very well for reliable transfer of elastic traffic. However, it is not applicable to multicast streaming since the way it is probing for available bandwidth produces rapidly varying transmission rates (sawtooth behavior). Its window-based control approach cannot meet the requirements of streaming video, which is better served by slowly-responsive and rate-based protocols that produce smoother transmission rates instead of mimicking TCP-behavior. A more fundamental problem of TCP is posed by its connection-oriented nature, which makes it unemployable for multicast.

Obviously, while necessary and powerful, TCP mechanisms are not sufficient to provide good services to streaming applications. Alternatively relying only on UDP without incorporating appropriate control mechanisms makes flows unresponsive to congestion indication and might lead to highly unfair situations. As shown by Floyd in [Flo00], unresponsive flows might easily starve TCP-based flows that still make over 90 % of the overall Internet traffic, as recent measurements indicate [FML+03]. In order to keep the Internet stable the IETF recommends alternative protocols to behave in a TCP-compatible way [BCC+98] and defines a TCP-compatible flow as:

> "... a flow that behaves under congestion like a flow produced by a conforming TCP. A TCP-compatible flow is responsive to congestion notification, and in steady-state it uses no more bandwidth than a conforming TCP running under comparable conditions (drop rate, RTT, MTU, etc.)."

The TCP-compatible paradigm transforms the requirement that all congestion control mechanisms must behave like TCP into a looser requirement that all congestion control schemes must be TCP-compatible. Thus, while TCP reacts to congestion signals within a round-trip time by halving the transmission window, a TCP-compatible flow can react "lazily" and smoothly, that is, on a larger time scale. In steady-state and over a larger time scale, however, it is required to use no more bandwidth than a conforming TCP flow running under comparable conditions. The cornerstone of this approach is the observation made by several research works that the bandwidth allocation of a TCP flow in steady-state can be well characterized by an analytical model.

A general consensus on the relative fairness between multicast and unicast traffic is still lacking, as discussed in Chapter 3. Hence, TCP-compatibility is considered a valid fairness definition accepted by the Internet community for unicast and multicast traffic. Consequently, we resort to this state-of-the-art fairness criterion throughout this thesis.

5.3 Background: TCP-Throughput Model

In recent years, there has been increasing interest in providing TCP-compatible solutions to rate-based control for unicast as well as multicast flows. Enabling tools for that purpose are analytical models that have been derived to capture the long-term steady-state throughput of TCP.

A simplified analytical model has been published in several forms [Flo91, LM97, MSMO97]. The model formulates the throughput T of a TCP flow as a function of the packet size s, a constant that is commonly set to $\sqrt{3/2}$, the round-trip time t_{RTT}, and the steady-state packet loss rate p:

$$T = \frac{s \cdot const}{t_{RTT} \cdot \sqrt{p}}. \tag{5.1}$$

The above model neglects TCP retransmission timeouts that become noticeable and more important at higher packet loss rates. To overcome this limitations, Padhye et al. in [PFTK98] and [Pad00] developed a more accurate model, which captures the throughput of a TCP flow according to:

$$T = \frac{s}{t_{RTT}\sqrt{\frac{2pn_{ack}}{3}} + t_{RTO} \cdot \min\left(1, 3\sqrt{\frac{3pn_{ack}}{8}}\right) p\left(1+32p^2\right)}, \tag{5.2}$$

where t_{RTT} denotes the round-trip time, t_{RTO} the retransmission timeout, n_{ack} the number of packets acknowledged by a received Acknowledgment (ACK), and p the packet loss rate.

A very prominent protocol utilizing the above model is TFRC. The latter has been extensively studied by the authors in [Pad00, FHPW00] and it has also been subject to a number of performance studies, for example [BBFS01] and [YKL01]. Meanwhile, it is in the IETF Standards Track [HFPW03] and has become part of the Datagram Congestion Control Protocol (DCCP) [KHF04, FKP04].

The solid and promising results obtained with the equation-based approach in the unicast case have motivated the use of the TCP-model also for multicast protocols. For example, TFRC has been extended to single-rate multicast in TCP-Friendly Multicast Congestion Control (TFMCC) [WH01], which recently has been utilized in Smooth Multi-rate Multicast Congestion Control (SMCC) [KB03] for multi-rate multicast. Both of the protocols rely on the original algorithms introduced by TFRC, in particular on the loss estimator.

In this thesis, the equation-based approach is employed for multi-rate multicast transmission scenarios. After having performed a number of preliminary testbed experiments with a modified implementation of TFRC, we implemented a framework in the network simulator *ns-2* [NS2] integrating and extending the original TFRC algorithms and mechanisms. Since we focus on streaming applications, we do not consider retransmission of lost packets. The value of the retransmission timeout t_{RTO} is approximated using the simple empirical heuristic $t_{RTO} = 4t_{RTT}$ that works reasonable well to provide fairness with TCP [FHPW00, HFPW03]. Furthermore, we set $b = 1$ according to the recommendation given by the IETF in [HFPW03].

Before we proceed with the discussion of the simulation results, we first summarize the key features of the model[1] particularly stressing the deployed loss mea-

---

[1] For a detailed description and the derivation of the model we refer the interested reader to [PFTK98] and [Pad00].

surement method. This discussion is crucial in order to understand the internals of the approach and appropriately interpret the observed behavior.

5.3.1 Loss Measurement

Congestion in the Internet is either signaled to the end systems implicitly by means of packet losses, or the routers explicitly notify the end hosts about impending congestion by marking packets. In the further discussion, we focus on the case of implicit signaling, which is still the predominant method. However, the algorithms can easily be extended to packet marking using Explicit Congestion Notification (ECN) mechanisms [Flo94, SA00, RFB01].

Equation 5.2 has been derived for the Reno flavor of TCP. It models the congestion avoidance behavior in terms of rounds, whereby a round starts with the back-to-back transmission of a congestion window size of packets. A round ends with the reception of the first ACK. Its duration equals a round-trip time and it is assumed to be independent of the window size. Furthermore, the time needed to send a full window of packets is observed to be smaller than the round trip time [Pax97a]. Packet losses in different rounds are assumed to be uncorrelated while losses in the same round are assumed to be correlated due to the back-to-back transmission. That is, if a packet is lost all remaining packets transmitted until the end of the round are considered lost as well.

TCP interprets lost packets as congestion signals and reacts to them by halving its congestion window. This is accounted for in the above model by computing the long-term TCP throughput for a given packet loss rate. Hence, an accurate estimation of the loss rate is crucial for the performance of the fair share estimator.

Loss Fraction

The obvious way to measure the loss rate is as a loss fraction calculated by dividing the number n_{lost} of packets that were lost by the number $n_{transmitted}$ of packets transmitted:

$$p_{fraction} = \frac{n_{lost}}{n_{transmitted}}. \tag{5.3}$$

In a receiver-driven approach suitable for multicast, each receiver can easily determine n_{lost} and $n_{transmitted}$ using the sequence numbers provided by the transport protocol over a certain time frame.

Loss Events and Loss Intervals

The loss fraction does not accurately model the behavior of widely-deployed TCP implementations [PF01] that employ the NewReno variant [FH99] or the Selective Acknowledgment (SACK) option [MMFR96]. Corresponding TCP flows halve the congestion window only once in response to one or several losses within one round-trip time. That is, losses in the same window of outstanding data are considered correlated and TCP treats them as a single *loss event*, as illustrated in Figure 5.2. The original algorithms of TFRC account for that behavior with a loss rate estimator that is based on loss events rather than packet losses.

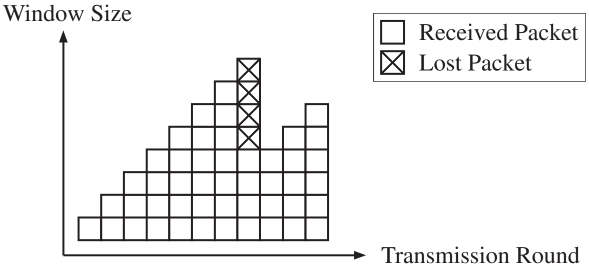


Figure 5.2: Schematic illustration of correlated losses for TCP.

TFRC being a rate-based protocol does not maintain a congestion window like TCP. To relate packets belonging to the same round, it uses the measured round-trip time as follows. On experiencing an initial packet loss at t_i a timer is triggered that ends after the round-trip time t_{RTT}. All packet losses until $t_i + t_{RTT}$ are considered belonging to the same loss event and are ignored. A new loss event can only be triggered if at least one round-trip time has elapsed since the beginning of the last loss event. Figure 5.3 schematically illustrates the principle.

The initial packet loss is followed by a period of time referred to as a Loss Insensitive Period (LIP) [WBB04]. Since a rate-controlled flow sends n_{RTT} packets per round-trip time, a LIP consists of $n_{RTT}-1$ packets. The number of packets sent between the initial losses of two consecutive loss events comprise a *loss interval* Θ_i. Thus by definition, a loss interval is always initiated and terminated by the initial packet losses that trigger two subsequent loss events. However, we consider packets sent following the most recent loss event being part of the actual non-terminated loss interval. Thus, the latter continuously grows until it is terminated by a new loss event.

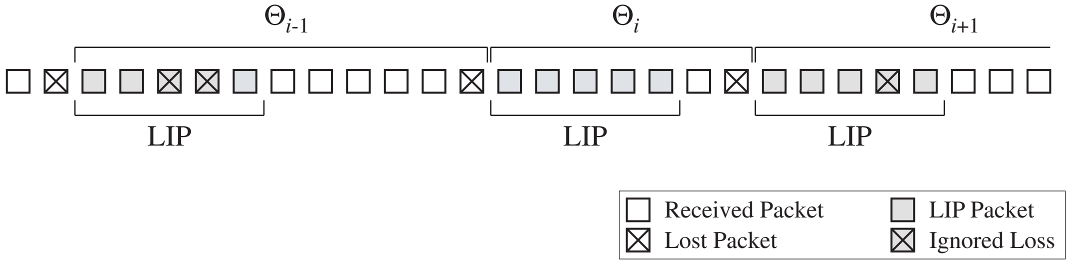


Figure 5.3: Schematic illustration of Loss Insensitive Periods (LIP) and loss intervals.

Generally, each packet sequence number is unambiguously assigned to a single loss interval. The packet that triggers a loss event might be either assigned to the interval it terminates or the following interval. However, for implementation reasons we decided to use the first realization, as illustrated in Figure 5.3.

Loss Event Rate

Having a model and definition for the loss event concept, the loss event rate can be straight-forwardly calculated using different methods. As a result of extensive testing and discussion of appropriate methods, Padhye et al. derived several guidelines for the unicast congestion control scenario [Pad00, FHPW00]:

- The loss rate for the TCP-throughput model should be calculated considering loss events rather than packet losses.
- The estimated loss event rate should track relatively smoothly and exhibit low variance in an environment with stable steady-state loss patterns.
- The estimated loss event rate should respond significantly to loss events occurring in several consecutive round-trip times.
- The estimated loss event rate should only decrease in two cases: in response to a new loss interval that is longer than the previously calculated interval, or when the actual non-terminated interval is long enough.

Since the above guidelines also hold for our work, we resort to the Weighted Average Loss Interval (WALI) method proposed by Padhye et al. WALI has provided best results in terms of the aforementioned guidelines compared to methods based on a dynamic window history as well as methods that use an exponentially weighted moving average of loss intervals [Pad00, FHPW00]. The use of a weighted average particularly reduces sudden changes in the estimated TCP-compatible rate that could result from unrepresentative loss intervals. Its excellent properties regarding smoothness and responsibility have made WALI the method of choice for equation-based control protocols also in the multicast domain [WH01, LLZ02, KB03].

The loss event rate according to the WALI method is computed using the actual non-terminated loss interval Θ_0 and the k most recent terminated loss intervals $\Theta_1, \ldots, \Theta_k$. Thereby, Θ_1 denotes the most recent terminated loss interval preceding Θ_0. It is important to ignore the non-terminated loss interval if it is short compared to the recent terminated intervals. However, if Θ_0 is sufficiently large, its impact should be taken into account, since otherwise the average loss rate will be overestimated. As a result, the average loss interval Θ_{avg} is computed as

$$\Theta_{avg} = \frac{\max\left\{\sum_{i=0}^{k-1} w_i \Theta_i, \sum_{i=1}^{k} w_{i-1} \Theta_i\right\}}{\sum_{i=0}^{k-1} w_i}, \tag{5.4}$$

where $w_0, \ldots, w_{k-1}$ denote the k weights assigned to the loss intervals. The loss event rate representing a measure of congestion is then calculated as the inverse of the average loss interval:

$$p_{event} = \frac{1}{\Theta_{avg}}. \tag{5.5}$$

Figure 5.4 shows the intersection of the history average loss interval and the most recent non-terminated loss interval. As the latter grows and becomes sufficiently large, it starts affecting the average loss interval and consequently the loss event rate.

The sensitivity to noise of the calculated loss rate and the responsiveness of the estimator are conflicting characteristics that depend on the history size k and the value of the weights $w_0, \ldots, w_{k-1}$. We follow the recommendations given in [WH01] for setting appropriate values and choose $k = 8$ and $w = \{5, 5, 5, 5, 4, 3, 2, 1\}$ since these settings are reported to provide a good trade-off between the sensitivity and responsiveness.

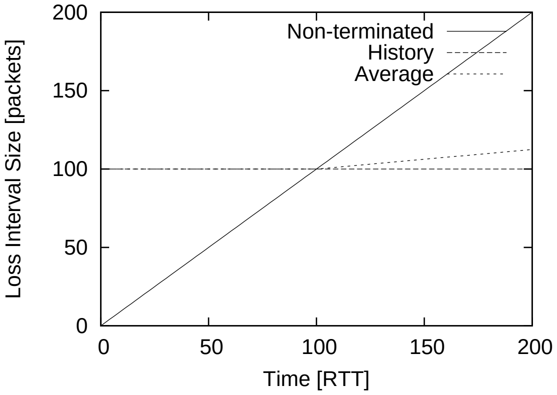


Figure 5.4: Progression of the average loss interval.

5.3.2 Loss Fraction versus Loss Event Rate

It is important to recall that in the original algorithm the impact of a loss event is independent of the actual number of packet losses within the loss event. That is, the latter represents a binary signal. If over a time interval only a single packet is lost in each loss event, then the sum of loss events equals the number of lost packets n_{lost}. Furthermore, the sum of loss intervals within the same time interval equals the number of transmitted packets $n_{transmitted}$. As a consequence, the packet loss rate in terms of loss events equals the loss fraction in Equation 5.3.

Nevertheless, if more than a single loss is experienced during the LIP of a loss event, the calculated loss rate is less than the actual packet loss fraction, which results in a weaker congestion signal. Assuming for the transmission channel a Bernoulli loss process with packet drop probability p_{drop}, a stream of n_{RTT} packets per round-trip time has the probability of experiencing one or more packet losses within a single round-trip time of:

$$\Pr(1 \leq X \leq n_{RTT}) = 1 - (1 - p_{drop})^{n_{RTT}}. \tag{5.6}$$

The loss event fraction is calculated as

$$p_{event} = \frac{1 - (1 - p_{drop})^{n_{RTT}}}{n_{RTT}}, \tag{5.7}$$

and decreases as the number n_{RTT} of packets per round-trip time increases. Thus, in the case of uncorrelated losses within a round-trip time, the loss event fraction p_{event} is inherently depending on the actual transmission rate. The effect is quite pronounced even for moderate packet loss rates, as Figure 5.5 illustrates. Uncorrelated losses are quite possible to be experienced in environments with a high degree of statistical multiplexing, or when routers implement Random Early Detection (RED) [FJ93] instead of drop-tail queue management.

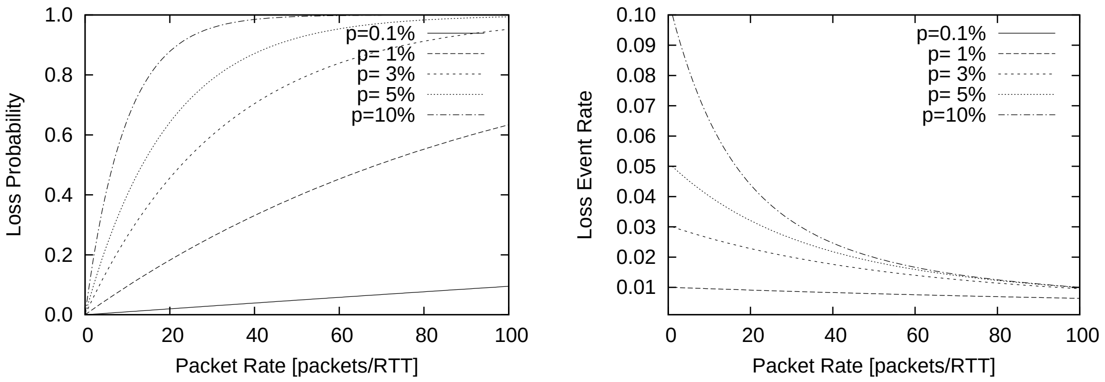


Figure 5.5: Loss probability and loss event rate as a function of the transmission rate for different packet drop probabilities.

Since the original algorithms have been developed in the context of unicast protocols that control the flow using a feedback loop, the transmission rate is determined by the control scheme, and so itself depends on p_{event}. Hence, the estimator performs reasonably well resulting in a smoothly varying transmission rate.

5.4 Applicability Evaluation

In contrast to their unicast counterparts, for a multicast source it is not feasible to maintain a tight control loop with every of its receivers due to scalability issues. Furthermore, under heterogeneous conditions it is obviously not possible for the source to adapt the transmission rates to the conditions of each individual receiver. This fundamental difference has to be considered when adopting the original algorithm.

However, to our knowledge this important issue has not been adequately addressed in existing work although it might severely distort the performance of the model, as we initially showed in [RSS03] and more thoroughly evaluate in the following.

5.4.1 Overview

After having introduced and analyzed the adopted approach, we used the experimental code of the TFRC implementation[2] and modified it in order to support open loop transmission. The results of the corresponding testbed experiments [Col04] approved our apprehensions and our initial simulation results presented in [RSS03]. Motivated by our findings we investigate the fair share estimator respectively its loss estimation algorithm in more detail. Since the preliminary testbed and simulation results correlated very well, the experiments have been performed using the network simulator *ns-2* [NS2].

The evaluation conducted within the current chapter utilizes the Adaptive Rate Control Framework (ARCF) that we developed and implemented in *ns-2* for investigating equation-based unicast and multicast control mechanisms. An overview of the design of the framework is given in Appendix C.

For the applicability study in this chapter, the algorithms and mechanisms of TFRC are implemented according to the IETF proposed standard [HFPW03]. The design of our framework allows to break up the original closed-loop control into an open-loop transmission with constant rate. This basically abstracts a multicast transmission scenario, where the transmission rate is not controlled and regulated to match the conditions of each receiver. The open-loop configuration allows to systematically study how different parameters such as the transmission rate and level of statistical multiplexing impact the estimator performance.

Recall that the fair share estimate is a function of the loss event rate, which is computed as the inverse of the average loss interval (see Equation 5.5). Since the latter depends on the underlying packet drop pattern, so does the estimated loss rate and the calculated TCP-compatible rate, respectively. The loss pattern is basically subject to the properties of the transport system and the traffic induced by the end systems. It depends on multiple parameters such as the network topology, the router mechanisms, the queue and link sizes, the end system control algorithms, etc.

In order to make the problem tractable, we follow a systematic simulation approach:

1. In a first set of experiments, an artificial channel is modeled exhibiting packet drops that are independent of the transmission rate of the flows. This usually

---

[2] The code is available at `http://www.icir.org/tfrc`.

holds for environments with a high level of statistical multiplexing. The loss and fair rate estimator should perform best under these conditions since the TCP model is based on the above assumption.

2. Following the artificial channel experiments, the rate estimation algorithm is analyzed under network conditions where flows compete directly for the resources of a bandwidth-limited bottleneck. Since the interesting characteristics to study is the TCP-compatibility, the parameter in the corresponding experiments is the number of competing TCP flows

3. The performance of the estimator as a function of the transmission rates is investigated for both of the above scenarios. That is, we study the impact of the packet rate over an artificial channel as well as a bandwidth-limited bottleneck in order to identify possible limitations.

4. A final experiment is intended to study how a recently proposed modification (*virtual packets*) to the estimation algorithm [WBB04] impacts the performance of the original algorithm.

Figure 5.6 gives an overview of the general evaluation methodology and Table 5.1 summarizes the simulation parameters and their used values.

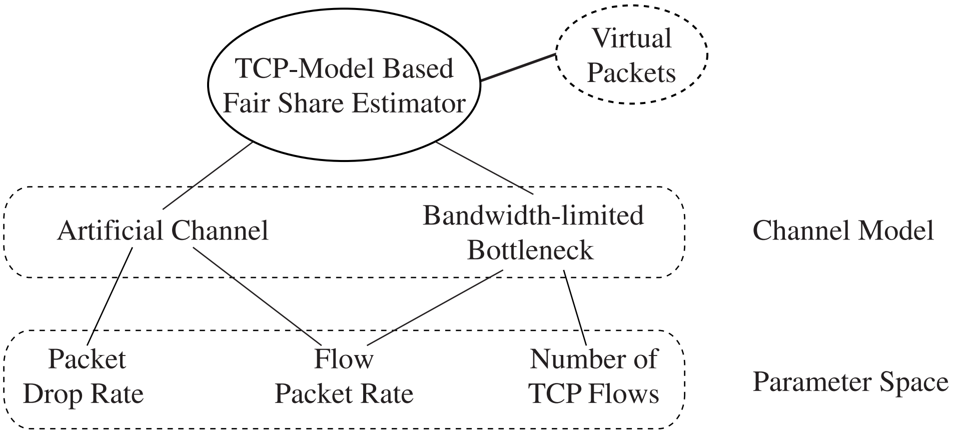


Figure 5.6: Overview of simulation methodology.

Parameter	Values
Packet Drop Probability [%]	0.1, 0.5, 1, 3, 5, 10
Number of competing TCP flows	1, 2, 4, 8, 16, 32
CBR Transmission Rate [packets/RTT]	1, 2, 4, 8, 16, 32, 64, 128

Table 5.1: Simulation parameters and values.

5.4.2 Simulation Configuration

In the simulation setup, one source-receiver pair is configured to instantiate the loss event rate and the fair share calculation module. Since the transmission rate of this flow is predetermined and kept constant over each simulation run, when discussing the scenarios this flow is denoted with CBR. A second source-receiver pair behaves corresponding to the specifications in [HFPW03]; thus, we refer to it as TFRC. Depending on the particular experiment, a single or multiple additional TCP instances are configured with the SACK option enabled.

For the simulation topology, we use a well-known double-star topology [BHH$^+$00] commonly referred to as the dumbbell topology and depicted in Figure 5.7. The sources and corresponding receivers are on either side of the bottleneck link. This abstract scenario is commonly used in unicast as well as multicast experiments in order to model the transport path from the source to the corresponding receiver with a single bottleneck link. Hence, modification of the simulation setup is easily performed by configuring path characteristics. Furthermore, when several flows share compete for the bottleneck resources, the scenario allows for direct comparison of fairness characteristics of the underlying protocols and control mechanisms.

For our experiments, access links are provisioned with 100 Mbps and a propagation delay of 5 ms. The bottleneck link between gateways G1 and G2 is configured with a 40 ms propagation delay, and either has a loss module inserted at gateway G1 (artificial channel) or a lower capacity (bandwidth-limited bottleneck).

All experiments have been conducted using equal packet size of $1,000$ Bytes. Simulation results are averaged and the confidence interval (95%) is computed over 15 runs for each parameter setting. We slightly varied the access link delays and the starting time of the flows in order to provide some degree of randomness and avoid phase effects [FJ92]. When discussing the rate calculated by the fair share estimator, we normalize the latter with the long-term average throughput achieved by the corresponding TCP flow.

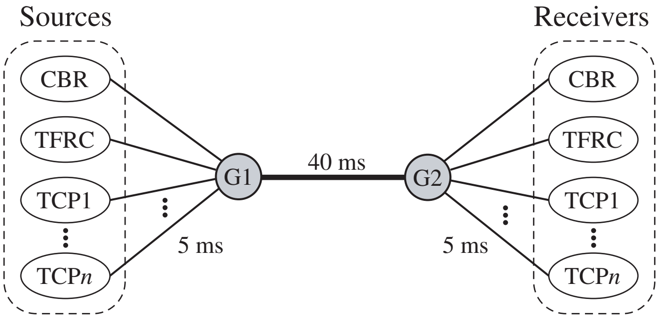


Figure 5.7: Dumbbell simulation topology.

5.4.3 Artificial Channel

Multiplexing is the process of forwarding packets from multiple sources to the same output link in a router. When this process is driven by demand of flows, it is performed on a statistical basis and referred to as statistical multiplexing. In routing systems where a large number of flows cross, the degree of statistical multiplexing is high. Emulating or simulating a high level of statistical multiplexing in an experimental environment is hardly feasible. However, since the relevant flows experience rather uncorrelated loss patterns that are assumed to be independent of the flow's transmission rate, we follow the common practice and resort to approximating this scenario with an artificial channel. Thus, we over-provision the bottleneck with 100 Mbps link capacity and insert a packet drop module to gateway G1 (see Figure 5.7). The packet drop probability is independent (memoryless) and varied in our experiments over the range[3] given in Table 5.1.

Figure 5.8 gives an overview of the loss event rate and the estimated fair share for different CBR transmission rates and packet loss probabilities. We observe that the estimated loss event rate follows the packet loss rate only for very low transmission rates. Already for a moderate loss probability of 3 % the loss event rate becomes underestimated for increasing transmission rates. This effect boosts up for higher loss rates and gives overestimated values for the fair share calculation.

In the case of a low packet rate and a low packet drop probability, consecutive packet losses are hardly experienced within a LIP. Thus, almost each packet loss triggers a new loss event and the loss event rate very closely follows the true packet loss rate. Under such conditions, the calculated fair share of the CBR flow almost

---

[3] The upper limit of 10 % packet loss probability has been chosen in accordance to the informal recommendations given in [BHH$^+$00] by members of the IETF Reliable Multicast Transport group.

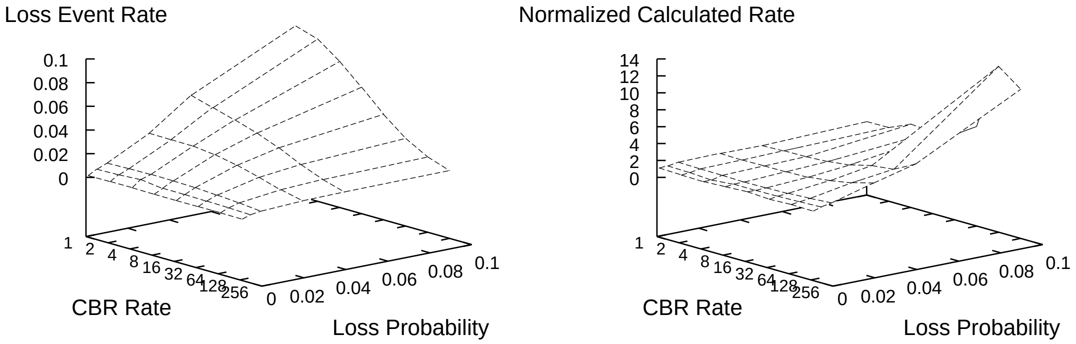
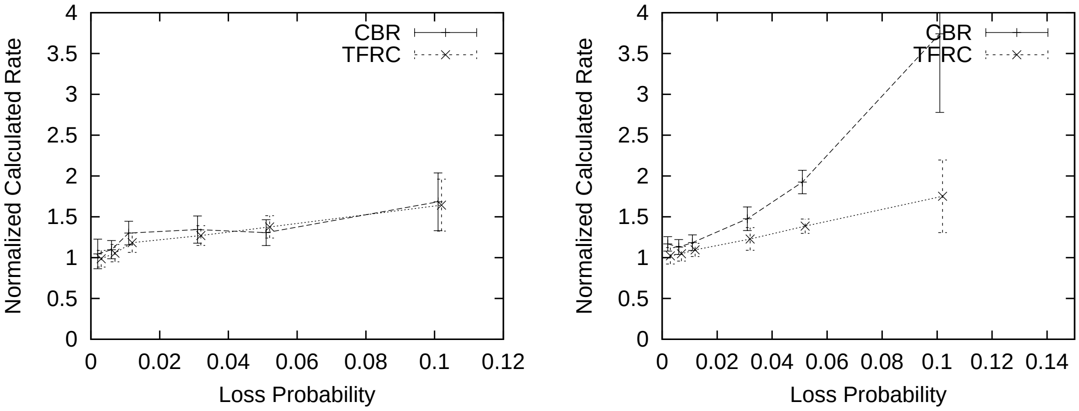


Figure 5.8: Overview of the loss event rate and calculated fair rate for the artificial channel scenario.

equals the rate calculated by a rate-controlled TFRC flow. This analytical discussion is approved by our simulation results presented in Figure 5.9(a) for a CBR flow with a transmission rate of one packet per round-trip time.



(a) 1 packet/RTT (b) 16 packets/RTT

Figure 5.9: TFRC rate and fair rate calculated by the CBR flow for two different transmission rates.

In the case of higher packet rates and a moderate packet drop probability, flows might experience multiple losses within a short time scale. If they occur within a time frame of a round-trip time, they are accounted for as a single loss event according to the LIP method. However, since losses on an artificial channel are uncorrelated, ignoring packet losses within the same LIP renders the loss event rate underestimated. As a result, the fair share will be overestimated as the simulation results depicted in Figure 5.9(b) demonstrate for a CBR flow with a transmission rate of 16 packets/RTT.

For further investigation of the impact of the transmission rate on the fair rate estimation, we varied the transmission rate of the CBR for fixed packet drop rates. Figure 5.10 illustrates the results obtained for an artificial channel where packets are dropped with a probability of 0.1 %. Due to the low level of losses, the CBR flow measures a loss event rate that very closely follows the loss event rate of the TFRC flow. Moreover, the loss event rate almost equals the actual loss rate so that both CBR and TFRC estimate a fair share that corresponds to the actual throughput achieved by TCP.

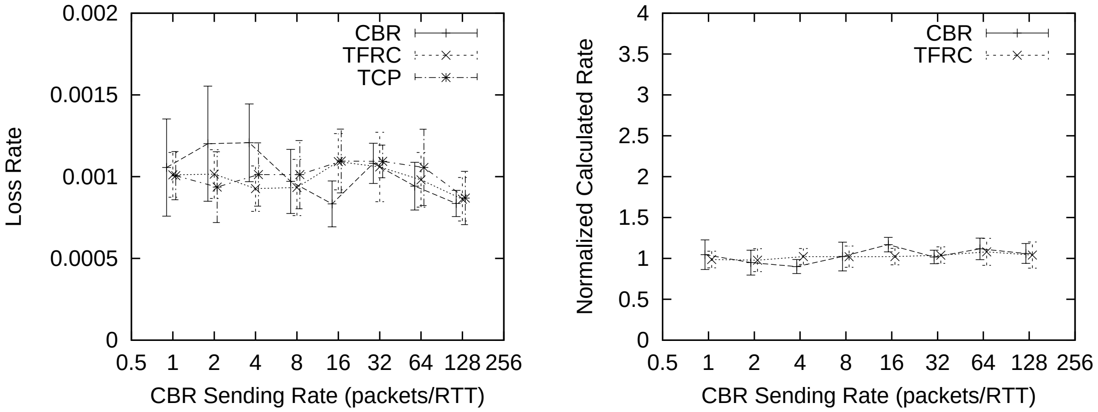


Figure 5.10: Measured loss rate and estimated fair share for an artificial channel with 0.1 % packet drop probability.

The results of the equivalent experiment with a moderate packet drop probability of 3 % is demonstrated in Figure 5.11. We observe that the loss estimation algorithm used with the open-loop CBR flow already becomes heavily dependent on the actual transmission rate of the flow. The measured throughput of the TCP connection in the experiment matches quite well the value of approximately 5.6 packets/RTT calculated using Equation 5.2. From the presented results it can be observed that for transmission rates up to the TCP-fair rate of 5.6 packets/RTT, the CBR and TFRC flow quite similarly measure a slightly underestimated loss rate. Thus, the calculated rate of both takes values of 1.2–1.5 times the actual TCP-equal rate. Since for this interval of transmission rates the CBR estimate very closely follows the TFRC estimate, the observed bias stems from the underlying method of the loss measurement rather than the absence of control mechanisms for the CBR transmission. However, as its transmission rate increases above the fair share, the CBR flow measures a decreasing loss rate, which leads to an overestimate of the fair share. This effect increases for higher loss rates and supports our observations previously presented in Figure 5.9(b).

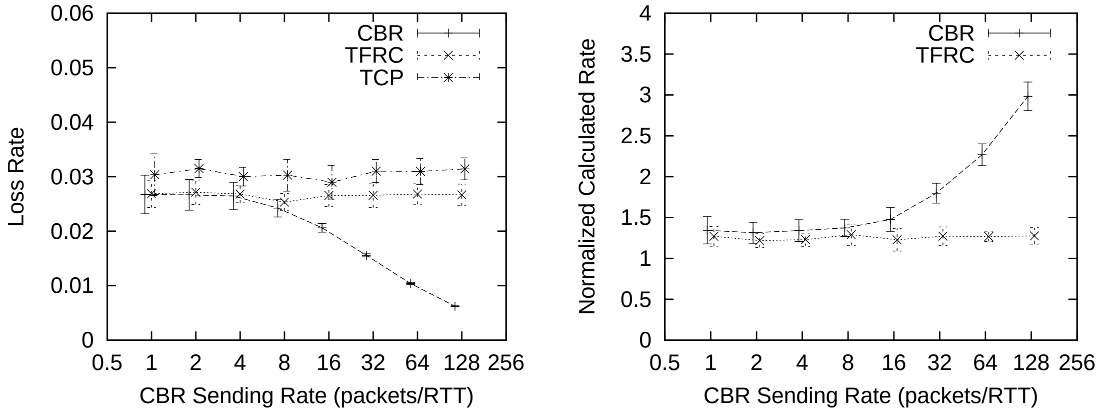


Figure 5.11: Measured loss rate and estimated fair share for an artificial channel with 3 % packet drop probability.

5.4.4 Bandwidth-Limited Bottleneck

When only a small number flows competes for the resources on a bandwidth-limited bottleneck, the assumption of uncorrelated packet losses made for the high-multiplexing scenario does not hold any more. The flows are very likely to interfere with each other and the loss patterns heavily depend on the employed control mechanisms and transmission rates of the flows. Hence, in the following experiments we study the characteristics of the rate estimation algorithm in a environment with a lower level of statistical multiplexing.

For the following experiments the level of statistical multiplexing is varied by configuring a number of n TCP flows to compete with the CBR and TFRC flows (see Table 5.1). The bottleneck capacity $C_{bottleneck}$ is set such that the theoretical fair rate of each flow is 16 packets/RTT, and the drop-tail bottleneck queue is dimensioned for buffering approximately 1.5 times the bandwidth-delay product. Since the bottleneck bandwidth is configured to $(n + 2) \cdot 16\,\text{packets/RTT}$, it effectively limits the number of reasonable simulation scenarios in the case of low values of n. For example, when $n = 1$, that is, there is only a single competing TCP flow, the bottleneck bandwidth is 48 packets/RTT. Running the CBR flow with a transmission rate of 128 packets/RTT would immediately overflow the bottleneck and produce results that are of no value. Hence, we run simulations only with transmission rates that do not exceed half of the bottleneck bandwidth.

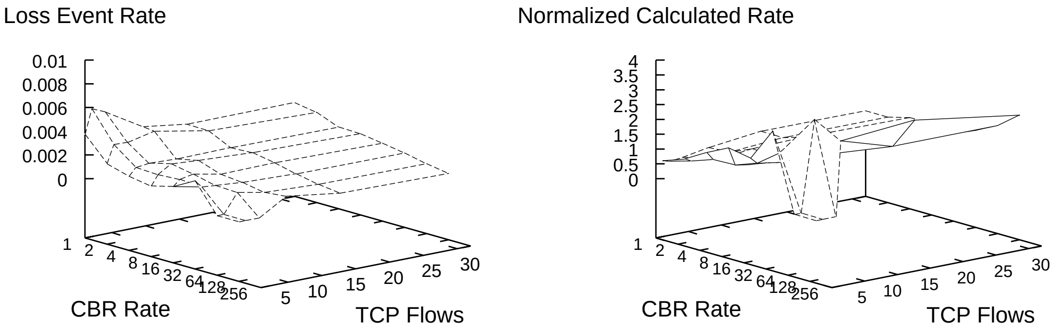


Figure 5.12: Overview of the loss event rate and calculated fair rate for the bandwidth-limited bottleneck scenario.

Figure 5.12 gives an overview of the loss event rate and the estimated fair share for the bandwidth-limited bottleneck as a function of the CBR transmission rate and the number of competing TCP flows. Although a dependency between the CBR transmission rate and the calculated loss event rate can be observed, the effect is much weaker compared to the results obtained for the artificial channel. The level of measured loss event rate is also at a very low level as the performed simulations show.

When congestion occurs in a router that multiplexes only very few competing flows, a CBR flow transmitting at a lower rate than its fair share measures a proportionally higher loss rate than the competing TFRC and TCP flows. The aggressive probing behavior particularly of TCP lets it consumes the available bandwidth and populate the queue at the bottleneck gateway. Recall that the bottleneck is dimensioned for a theoretical fair share of 16 packets/RTT for each flow. Thus, as the CBR transmits at a comparably low rate, inter-packet spacing is large and single losses have a high impact on the loss estimation algorithm. As a result, the congestion state is overestimated leading to a very conservative behavior of the rate estimation algorithm for this scenario.

The simulation results depicted in Figure 5.13 show how the measured loss event rate of a CBR flow transmitting at 1 packet/RTT changes as the number of competing TCP flows varies. For a very low degree of multiplexing it is too sensitive to losses leading to an underestimated fair share that equals approximately 50 % of the TCP throughput. In contrast, TFRC utilizing the same TCP-model is even more aggressive than TCP due to the employed rate control mechanism. As the multiplexing level increases, the calculated rate of CBR converges to the actual throughput of the TCP flow.

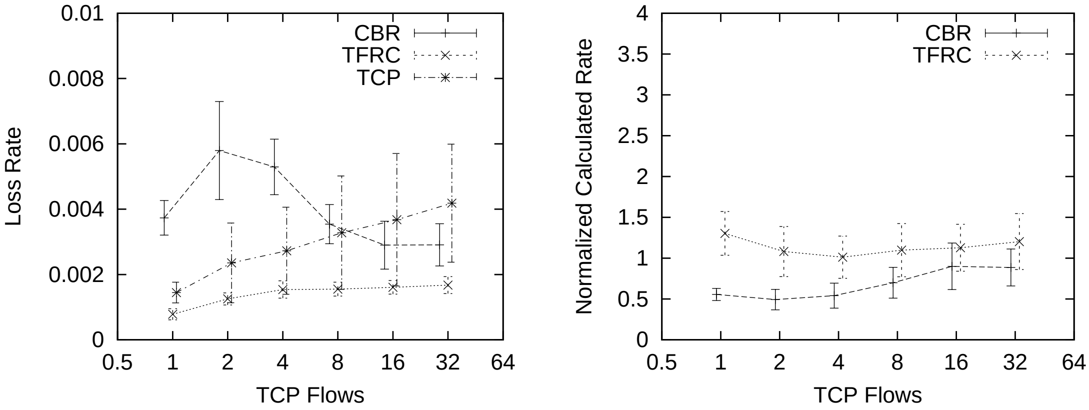


Figure 5.13: CBR flow transmitting 1 packet/RTT and sharing a bandwidth-limited bottleneck with a TFRC and a varying number of TCP flows.

In contrast to the above scenario, when sending at rate that is above its fair share, a CBR flow competing with other rate-controlled flows tends to underestimate the loss rate and calculates a fair rate that is too high. Figure 5.14 demonstrates this behavior for the CBR flow configured to transmit at a constant rate of 32 packets/RTT, which is two times the theoretical fair rate. Since the loss event rate measured by the CBR receiver is significantly lower than the loss rate experienced by the TCP flow(s), the calculated rate does not converge to the TCP-fair value even for a higher multiplexing level. Actually, the simulation results indicate that as the multiplexing is further increased so is the congestion level. Thus, the estimated loss event rate decreases and the calculated rate does not approach a fair level. This behavior correlates quite well with the observations discussed for the artificial channel in Section 5.4.3.

Corresponding to the evaluation procedure in the case of an artificial channel, the impact of the transmission rate on the fair rate estimation has been further investigated. Therefore, the CBR transmission rate was varied for a fixed number of competing flows. Figure 5.15 illustrates the results of the CBR flow sharing a bandwidth-limited bottleneck with a TFRC flow and a single TCP flow. We observe

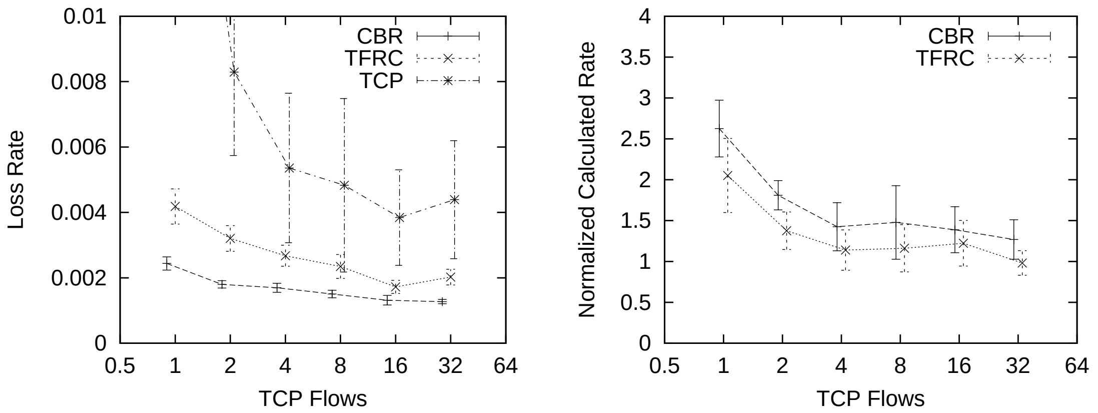


Figure 5.14: CBR flow transmitting 16 packets/RTT and sharing a bandwidth-limited bottleneck with a TFRC and a varying number of TCP flows.

that at a transmission rate of 1 packet/RTT the CBR flow measures a higher loss rate than both TFRC and TCP. As the CBR flow increases its transmission rate, its estimated loss event rate decreases very fast to values below that of TCP. Recall that the bottleneck link is configured for a theoretical fair share of 16 packets/RTT. However, the calculated fair share of the CBR overshoots the TCP-fair level already at a transmission rate of approximately 4 packets/RTT. The TFRC flow generally allocates more bandwidth than the TCP flow.

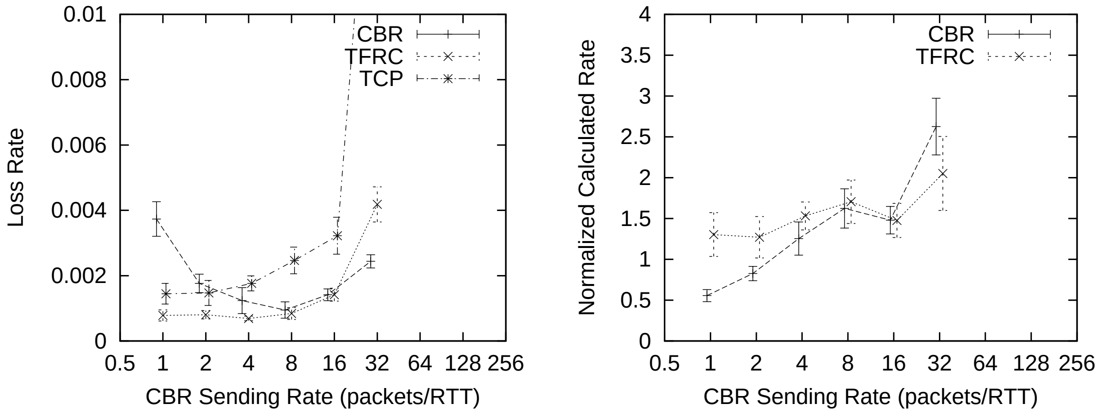


Figure 5.15: Measured loss rate and estimated fair rate of a CBR and TFRC flow sharing a bandwidth-limited bottleneck with a TCP flow.

The results of the equivalent experiment with the CBR and TFRC flow sharing the bottleneck with 8 TCP flows is presented in Figure 5.16. Although the loss overestimation is still observed at very low transmission rates, it is less pronounced and the estimated fair rate converges with that of TFRC as the transmission rate of the CBR flow increases. However, as the transmission rate exceeds the fair share of approximately 16 packets/RTT the fair rate becomes overestimated.

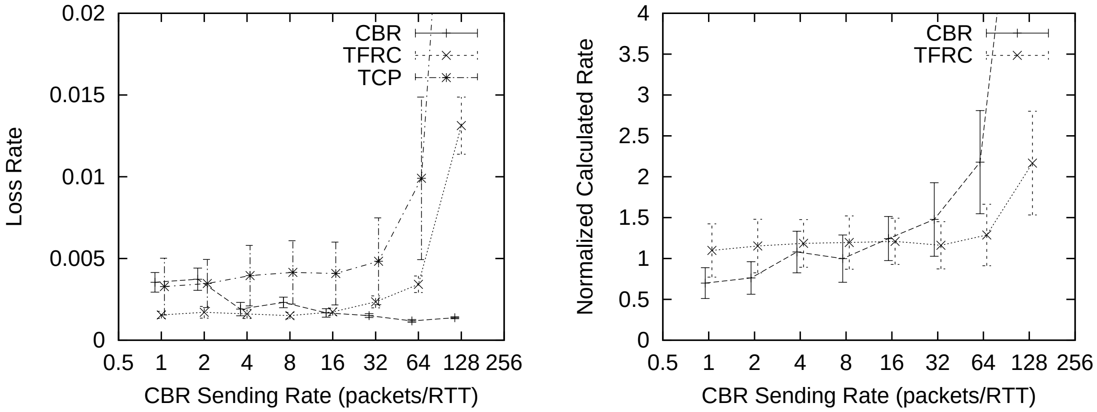


Figure 5.16: Measured loss rate and estimated fair rate of a CBR and TFRC flow sharing a bandwidth-limited bottleneck with 8 TCP flows.

5.4.5 Virtual Packets

Only recently Widmer et al. in [WBB04] very thoroughly studied the TCP model for applications that perform rate control by adapting the packet size instead of the packet rate. The results indicate that in the case of variable packet size the performance of the fair share estimator intrinsically depends on the packet size. The authors proposed a modification to the loss event rate calculation that addresses this problem, and they assumed their algorithm to also solve the limitations of the basic rate estimator that we identified in our extensive evaluation.

For a constant data rate r_{const} the packet rate is a function of the packet size. If a source sends n'_{RTT} packets with size s' per round-trip time, to achieve the same bit rate with a smaller packet size $s'' < s'$ it obviously has to increase the packet rate:

$$R = n'_{RTT} \cdot s' = n''_{RTT} \cdot s'' \qquad \Rightarrow \qquad n''_{RTT} = n'_{RTT} \frac{s'}{s''}. \tag{5.8}$$

As Widmer et al. showed in [WBB04], in an operating regime with smaller packet size and constant data rate the increase in the number of loss events is no longer proportional to the increase in the number of packets the loss events are sampled over. The origin of this effect can be explained referring to the analytical results of Ramesh and Rhee presented in [RR99].

Assuming a Bernoulli packet loss process as for the artificial channel in Section 5.4.3, the expected value for a loss interval in steady state is:

$$E(\Theta) = n_{RTT} - 1 + \frac{1}{p_{drop}}, \tag{5.9}$$

where $n_{RTT} - 1$ is the number of packets sent within a LIP following a packet loss that triggers a loss event (see Section 5.3.1), and p_{drop} denotes the packet drop probability. Obviously, the higher the number of packets per round-trip time n_{RTT}, the larger is the expected loss interval. Hence, the loss event rate is clearly depending on the packet rate. This fundamental and general finding is important for the purpose of dealing with variable packet sizes, but helps also for the understanding of some of the effects observed in our experiments.

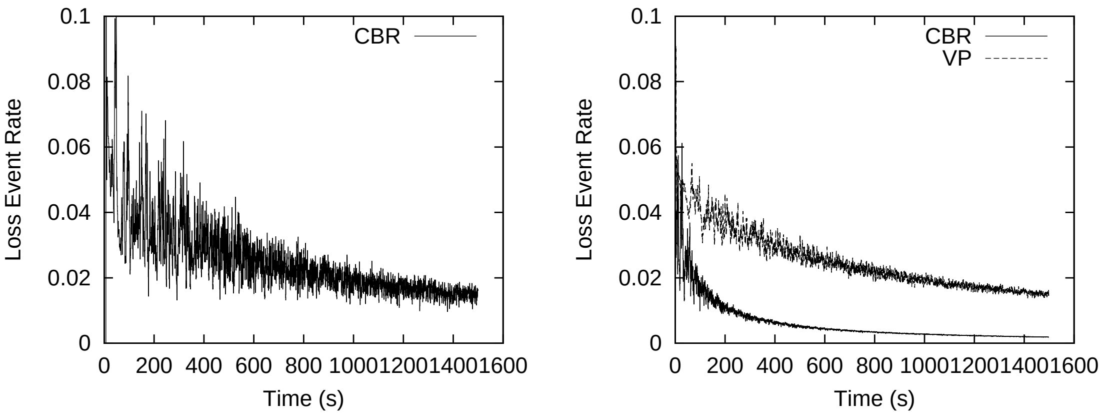
To cancel the bias in favor of flows with smaller packets, Widmer et al. introduced the concept of virtual packets. The main idea of this method is to combine smaller packets of size s into virtual packets of reference size S. Given that all other parameters remain the same, flows employing the virtual packet method for the estimation of the loss event rate should calculate a throughput that is independent of the packet size.

In order to verify whether the method of virtual packets is capable of reducing or eliminating the limitations of the loss estimation algorithm, we conducted simulations using an artificial channel with a packet drop probability of 5 %. The actual packet size is set to (a) 1,000 Bytes respectively (b) 100 Bytes; the virtual packet size is 1,000 Bytes. The transmission rate of the CBR flow is increased over time. Figure 5.17 shows the results of one simulation run.

The trace of the measured loss event rate (without virtual packets) in Figure 5.17(a) clearly shows a dependency on the transmission rate. This corresponds to our findings in the preceding sections. Figure 5.17(b) depicts the loss event rate calculated by the same flow with packets of size 100 Bytes (lower trace). According to the analytical study, the use of smaller packets lead to an underestimation of the loss rate. When applying the virtual packet method, the CBR flow measures a loss event rate that corresponds[4] to that measured without virtual packets and

---

[4] Note that the trace for the flow with virtual packets is smoother than the trace for the flow with actual packet size of 1,000 Bytes. This can be attributed to the finer sampling of losses when



(a) 1,000 Bytes packet size (b) 100 Bytes packet size

Figure 5.17: Loss event rate measured over an artificial channel with a packet drop probability of 5 %.

packet size 1,000 Bytes, as depicted in the upper trace of Figure 5.17(b).

The results of the performed simulations support that the virtual packet method proposed in [WBB04] can cancel the bias in the calculated fair share caused by variable packet sizes. However, the influence of the transmission rate on the loss event rate estimator as demonstrated in this thesis still remains. This is easily validated by analysis using Equation 5.9. If we substitute the number of real packets n_{RTT} with the number of virtual packets $n_{virtual}$, the expected loss interval is calculated as a function of the transmitted virtual packets. As the actual packet rate increases, so does the virtual packet rate due to their linear relation. Hence, the virtual packet method does not reduce the influence of the transmission rate on the fair share calculated using the TCP-throughput model.

5.4.6 Impact on Multi-rate Multicast

The results of our simulations confirm that the state-of-the-art equation-based approach to rate control performs reasonably well over a wide range of parameter setting for open-loop transmission scenarios. Hence, the underlying rate estimation algorithm is a powerful tool that can be also used by multicast receivers participating in a multicast session. However, there are two pathological cases that have been identified and analyzed:

---

using smaller packets.

1. When a decoupled[5] flow in environments with moderate to high packet loss rate sends data at a higher rate than the fair share, the corresponding receiver tends to overestimate the fair share. The overestimation effect is boosted as the transmission rate and loss rate increases. As a consequence, receivers participating in a multi-rate multicast session might report an overestimated TCP-compatible rate and choose a subscription level that is highly unfair.

2. When a decoupled flow shares a bandwidth-limited bottleneck with rate-controlled congestion-controlled TCP traffic, the estimated fair rate of a corresponding receiver heavily depends on the transmission rate and level of multiplexing. Similarly to the behavior in environments with uncorrelated and high loss rates, the estimator tends to overestimate its fair rate for transmission rates that exceed the TCP-fair rate. Basically, this effect supports the findings in [BCC$^+$98]. On the other hand, for lower transmission rates and a very low number of competing TCP flows, the loss rate is overestimated so that the calculated rate is lower than the actual fair TCP-compatible rate. Transferring this effect to the multi-rate multicast scenario, a receiver that performs group subscription based on the fair rate estimated might behave very conservatively. While this does not harm the cross traffic, it might lead to unfair situations for multicast receivers.

5.5 Loss Measurement Modifications

This section introduces modifications to the basic loss measurement method in order to improve the fair share estimation using the TCP model under the conditions of open-loop transmission. Therefore, we compare the assumptions made for the TCP model to the operational conditions of a multicast streaming flow. Based on the insight gained through analysis and experiments a new concept of *loss event impact* is introduced and an corresponding loss event rate algorithm is devised.

---

[5] We use this term to refer to a flow that in contrast to TCP and TFRC flows is not subject to closed-loop rate-control. Thus, it sends at a predetermined rate and neither reacts to congestion signals nor probes for available rates. This abstracts a multicast scenario where the transmission rate cannot be adjusted to the bandwidth capabilities of each receiver.

5.5.1 Model Assumptions

Packet losses are implicit signals of congestion on an end-to-end path. Recall from Section 5.3.1 that the TCP-model underlying the equation-based fair share estimator assumes that packet losses experienced within a round-trip time (round) are correlated (see Figure 5.2). On the other hand, losses from different rounds are assumed to be uncorrelated. The rationale behind this is that TCP is assumed to send packets back-to-back in each round. When a packet burst arrives at a drop-tail queue with insufficient buffer space, one or more packets of that transmission round will be dropped. Prominent TCP variants react to packet loss by halving their congestion window only once per round-trip time regardless of the actual number of lost packets. The loss estimation algorithm used in the equation-based approach accounts for that behavior by ignoring all detected losses following an initial packet drop that triggers a loss event. Thus, the impact of losses within a round-trip time is constant and independent of the actual number of dropped packets.

The equation-based approach has proved to be well-suited for flows using closed-loop control to regulate their transmission rate although the loss estimator is inherently influenced by the operational rate (see Section 5.3) However, the simulation results presented in Section 5.4 showed that pathological cases might occur when opening up the control loop. Particularly, in environments with steady-state losses a decoupled flow transmitting at rates higher than a greedy TCP flow tend to overestimate its fair share. It is observed that while the packet rate increases the estimated congestion level decreases.

The bias in loss estimation basically originates from two facts: (a) at a steady-state packet loss rate, the expected value of the loss interval according Equation 5.9 is intrinsically a function of the transmission rate (see Section 5.4.5); (b) a rate-based flow in contrast to its window-based counterpart does not send a burst of packets back-to-back but rather evenly-spaced, so that losses occurring in the same round-trip time are not necessarily correlated. This assumption is confirmed by the simulation results presented in the previous section.

5.5.2 Loss Event Impact

In order to improve to performance of the loss event rate estimator, we extended the concept of loss events. While a loss event in the original algorithm is basically a binary congestion indicator, in our modification it is also assigned a value for its impact. We define the *loss event impact* of the ith loss event as:

$$\Psi_i = (n_{lost}[i])^{1-\gamma} \qquad \text{for} \quad \gamma \in [0..1], \tag{5.10}$$

where $n_{lost}[i]$ denotes the number of packets that are lost within the ith loss event, and γ is a loss correlation factor. Since with the above definition all packet losses can be considered in the loss event rate calculation, the Loss Insensitive Period (LIP) is substituted with a Loss Aggregation Period (LAP).

Having defined the loss event impact, the average value over the k recent loss event impact values $\Psi_0, \ldots, \Psi_{k-1}$ is then calculated as follows:

$$\Psi_{avg} = \frac{\sum_{i=0}^{k-1} w_i \Psi_i}{\sum_{i=0}^{k-1} w_i}, \tag{5.11}$$

where Ψ_0 corresponds to the most recent loss event, and the weights $w_0, \ldots, w_{k-1}$ are the same as used for calculating the average loss interval (see Section 5.3.1).

Finally, the new loss event rate is computed as the ratio of the average loss impact and the average loss interval (Equation 5.4):

$$p_{event} = \frac{\Psi_{avg}}{\Theta_{avg}}. \tag{5.12}$$

The loss correlation factor γ in Equation 5.10 is generally depending on the actual loss pattern. As its name already implies, the factor provides a mean to parameterize the probability of two losses experienced in the same round being correlated. If the $\gamma = 1$, every loss event has equal impact of 1. In that case, all losses in a round-trip time are considered correlated, which corresponds to the original algorithm using LIP. On the other hand, if γ is set to values close to 0, the impact of a loss event heavily depends on the number of actually lost packets. Since we focus on rate-based transmission for streaming media, experienced losses are assumed to be rather uncorrelated and the event impact factor is heuristically set to $\gamma = 0$.

In the following section, we present results obtained in experiments using the modified loss event rate estimator based on loss event impacts. They clearly show how the performance of the loss estimation improves for the decoupled open-loop flow.

5.6 Evaluation of the Modified Algorithm

To evaluate the modified loss rate estimator and compare it to the original algorithm, in this section the same simulation configuration and setup is used as in Section 5.4. Moreover, both section are similarly structured in order to make fast comparison possible.

In the following experiments, we the original equation-based rate estimator in the CBR flow is replaced by our modified estimator. When discussing the results, the new CBR flow is referred to as CBR-LAP while the flow instantiating the original algorithm is denoted with CBR-LIP.

5.6.1 Artificial Channel

Figure 5.18 gives an overview of the loss event rate and the estimated fair rate over the CBR transmission rate and the packet loss probability of the artificial channel. It clearly shows that the relationship between the measured loss event rate and the packet drop probability of the channel is almost independent of the CBR-LAP transmission rate. Moreover, the calculated fair share maintains a highly TCP-compatible level over the wide range of the parameter settings.

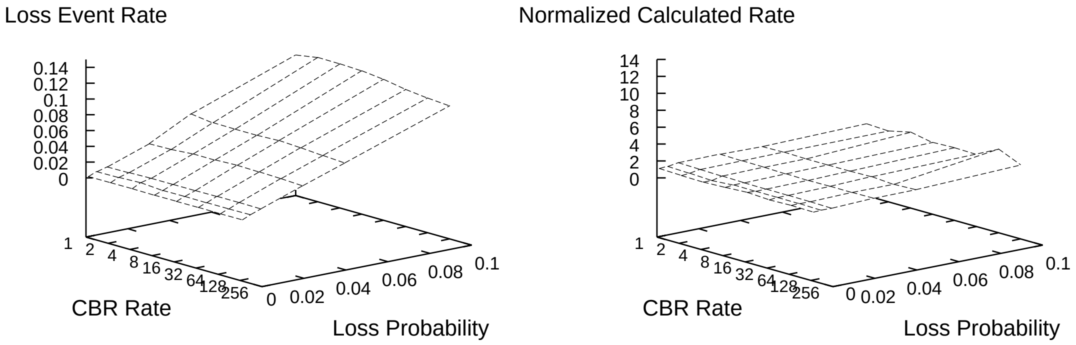
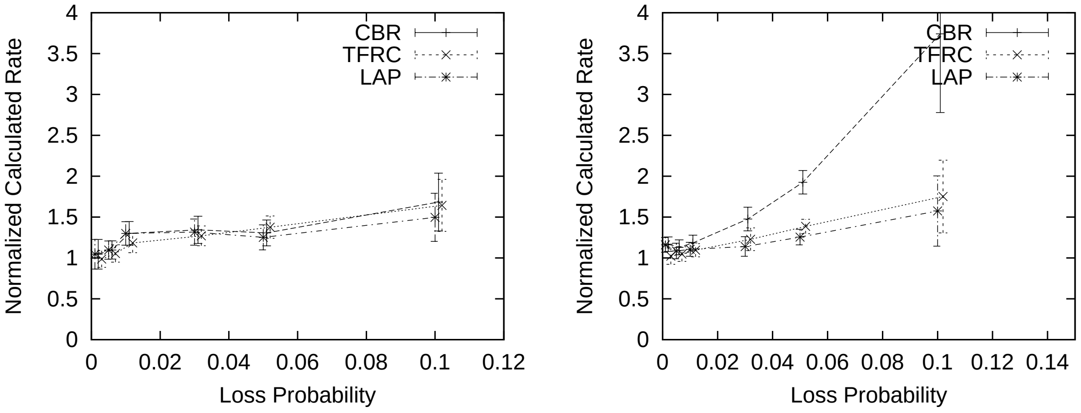


Figure 5.18: Overview of the loss event rate and calculated fair rate of the CBR-LAP flow in an artificial channel scenario.

Equivalent to its LIP counterpart, at low packet rates and low packet drop probabilities the CBR-LAP flow does hardly experience two or more packet losses within the same round-trip time. Consequently, the impact of a loss event is constantly 1 and the behavior corresponds to that of the LIP algorithm. This is evident in Figure 5.19(a) for the CBR flow transmitting at 1 packet/RTT, which is comparable to the results of the CBR-LIP flow in Figure 5.9(a).



(a) 1 packet/RTT (b) 16 packets/RTT

Figure 5.19: TFRC rate and fair rate calculated by the CBR-LAP and CBR-LIP flow, respectively, for two different transmission rates.

However, in the case of higher transmission rates, the LAP algorithm clearly outperforms the LIP algorithm, as Figure 5.19(b) exemplifies for a transmission rate of 16 packets/RTT. When multiple packets are sent per round-trip time in a high-multiplexing environment, the probability of experiencing uncorrelated losses within the same loss event raises. As the loss probability takes a moderate to high level, the LIP algorithm increases to underestimate the congestion state while LAP closely follows the behavior of the closed-loop TFRC algorithm.

For further investigation of the relationship between the transmission rate and the estimation performance, in the following simulations we varied the transmission rate while keeping the packet drop probability of the channel constant.

Figure 5.20 shows the results obtained for a packet dropping probability of 0.1 %. Packet losses are very rare so that each loss event consists of a single loss, and the loss interval becomes very large. Furthermore, the second term of Equation 5.9 for the expected value of the interval (see Section 5.4.5 becomes dominant, so that the number of transmitted packets per round-trip time has hardly any influence. As a result, over a wide range of the CBR transmission rate the performance of all three equation-based algorithms (TFRC, LIP and LAP) is not distinguishable, and the estimated fair level very closely follows the actual TCP-fair rate.

In contrast, when the drop probability is increased, the difference between the LIP and LAP performance becomes evident. The simulation results obtained for

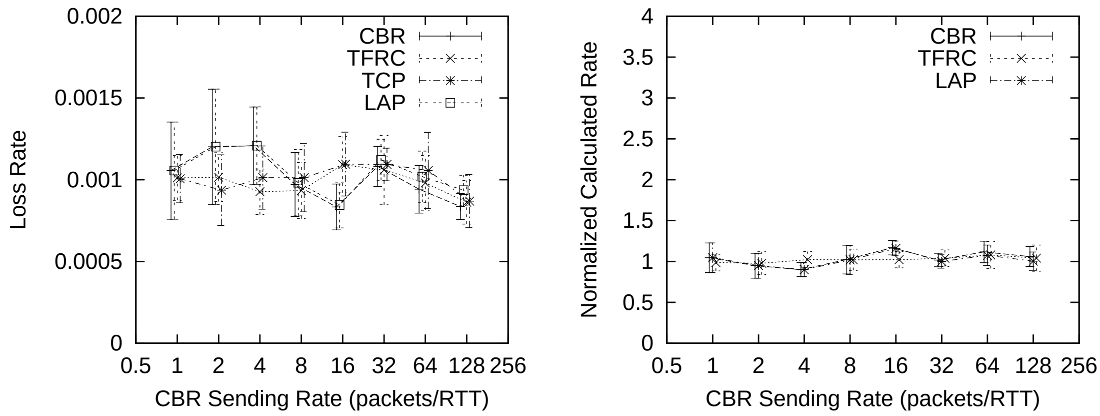


Figure 5.20: Measured loss rate and estimated fair share for an artificial channel with 0.1 % packet drop probability.

an environment governed by 3 % packet drop probability is demonstrated in Figure 5.21. On average every 33rd packet is dropped under these conditions, so that a divergence between LIP and LAP is expected to be noticeable at packet rates of more than 33 packets per round-trip time. However, we observe that the LIP algorithm starts to underestimate the loss rate already at packet rates that are slightly above the TCP-compatible rate of 5.6 packets/RTT leading to fair share overestimation. As the transmission rate increases this effect boosts. In contrast, our LAP approach performs fairly over the complete range of configured transmission rates and maintains a fairness level closely following that of the TFRC flow.

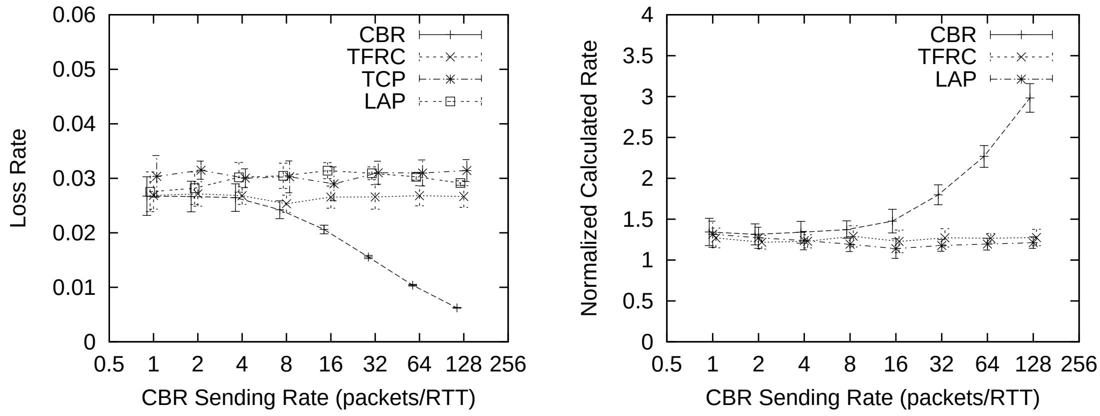


Figure 5.21: Measured loss rate and estimated fair share for an artificial channel with 3 % packet drop probability.

5.6.2 Bandwidth-Limited Bottleneck

An overview of the performance of the LAP approach using loss event impacts under the condition of directly competing flows on a bandwidth-limited bottleneck link is presented in Figure 5.22.

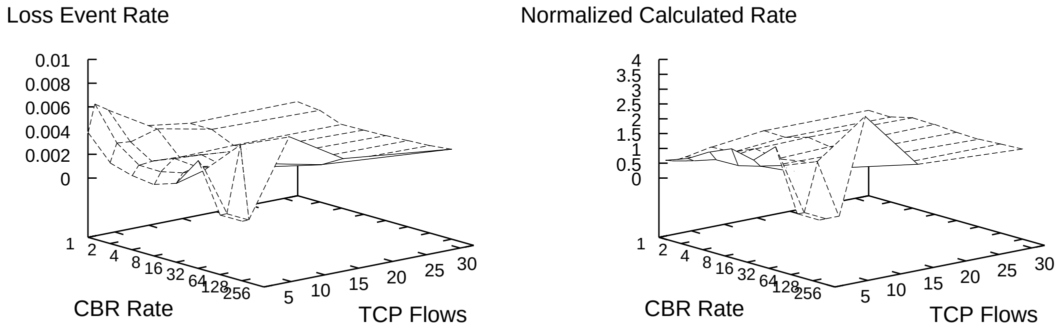


Figure 5.22: Overview of the loss event rate and calculated fair rate for the bandwidth-limited bottleneck scenario.

The graphs visualize that at higher degrees of statistical multiplexing the CBR-LAP obtains a quite fair estimate of the loss rate and the fair share. However, when only few flows are competing for the bottleneck resources and the CBR-LAP flow transmits at low packet rates, it tends to overestimate the loss rate. In this situation, the fair share is underestimated. Recall that the bottleneck bandwidth in this simulation setup is configured to $(n + 2) \cdot 16$ packets/RTT, where n denotes the number of competing flows. Since we limited the transmission rate of the CBR flow to at most half of the bottleneck bandwidth, values in the graph for higher transmission rates have not been obtained and have been manually set to zero. This explains the sudden slope as the number of TCP flows approaches 1 concurrently with the packet rate increasing to very high values.

Figure 5.23 depicts the results of the simulation configuration when the CBR sends 1 packet/RTT and sharing the bottleneck with a TFRC and one or more TCP flows. Due to the very low packet rate, the loss measurement process employed in LAP basically corresponds to the original LIP approach. LAP measures a similar loss rate and fair share as LIP, and suffers the same limitations. When only few flows share the bottleneck, the transmission behavior of each flows directly impacts the experienced loss pattern. The measured loss rate of a flow then depends on its own transmission behavior and that of the other flows. Since in contrast to TCP and TFRC the CBR flow does not aggressively probe for available bandwidth,

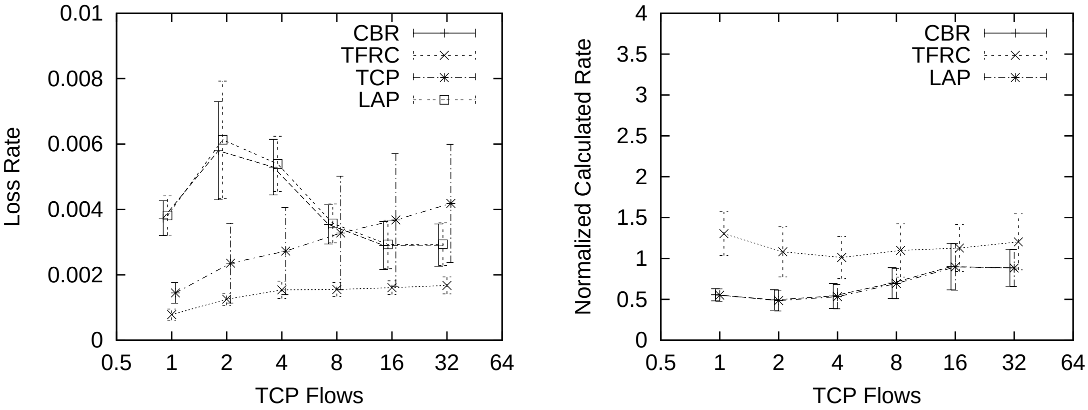


Figure 5.23: CBR flow transmitting 1 packet/RTT and sharing a bandwidth-limited bottleneck with a TFRC and a varying number of TCP flows.

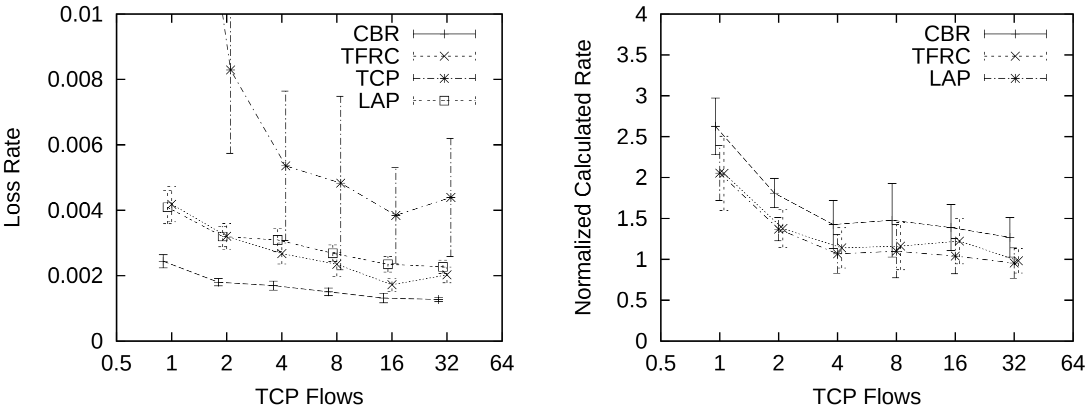


Figure 5.24: CBR flow transmitting 16 packets/RTT and sharing a bandwidth-limited bottleneck with a TFRC and a varying number of TCP flows.

it underestimates the fair share. As the number of competing flows increases the calculated rate approaches a fair level. The reason is obvious: the multiplexing level increases so that the influence of the flow behavior on the loss process diminishes. Hence, the CBR flow faces a similar solutions as in the artificial channel simulations.

For high transmission rates that overshoot the fair share, Figure 5.24 shows a somewhat different behavior. While for low transmission rates overestimating the loss rate and calculating a very conservative fair share, LIP and LAP turn to be too aggressive when packets are sent at high rates. This can be attributed to the aforementioned reason, that the characteristics of the flows have sever impact on the experienced loss process.

Finally, Figure 5.25 and Figure 5.26 demonstrate the performance of the LAP approach over a wide range of packet rates when the CBR and TFRC flow are sharing the bottleneck with a single TCP flow and with eight TCP flows, respectively. Interestingly, LAP follows LIP as long for packet rates below the fair share, while for increasing packet rates the LAP estimator follows TFRC. In this operating regime, LAP maintains a relatively fair estimate and outperforms the LIP algorithm.

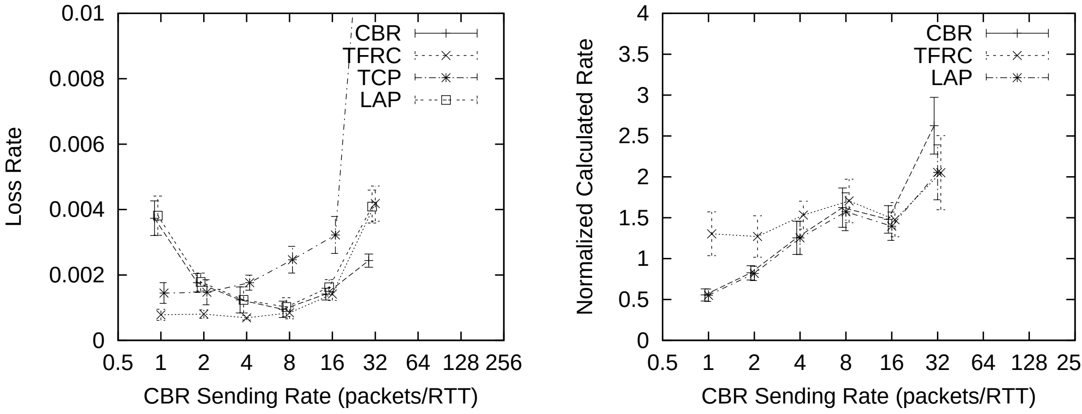


Figure 5.25: Measured loss rate and estimated fair rate of a CBR-LAP and TFRC flow sharing a bandwidth-limited bottleneck with a TCP flow.

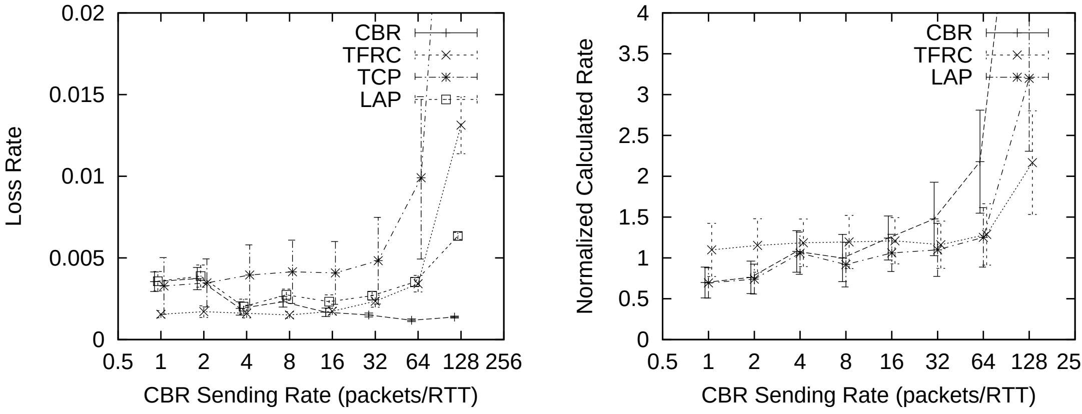


Figure 5.26: Measured loss rate and estimated fair rate of a CBR-LAP and TFRC flow sharing a bandwidth-limited bottleneck with 8 TCP flows.

5.6.3 Discussion

The results of our experiments verified that the loss estimator modifications presented in Section 5.5 can improve the equation-based fair share estimator. This holds particularly for environments of high-statistical multiplexing, where the original method (LIP) tends to overestimate the fair share when the corresponding flow sends at a higher packet rate than a greedy TCP flow. Since the modified algorithm (LAP) assigns a measure of impact to each loss event rate, it has been shown that it is much better suited to accurately measure the congestion under extreme conditions. The estimator becomes independent from the actual transmission rate.

The latter is an important feature for equation-based layered multicast sessions where the cumulative rates of a low number of layers obviously cannot accommodate heterogeneous end-to-end path conditions. Thus, participating receivers might overestimate their fair share, which when used for stream optimization (see Chapter 3) will lead to reduced optimization performance. Furthermore, an overestimate of the fair share might force receivers to join a higher transmission rate, which then further degrades the precision of the estimator and so forth.

While the LAP method can improve the loss estimation when flows face the aforementioned conditions, it suffers the same limitations as the LIP algorithm at a very low degree of statistical multiplexing and drop-tail queues. When only few flows share the bottleneck, the transmission behavior of each flow directly impacts the experienced loss pattern. In this situation, the different flows might actually measure different loss patterns depending on their own and the characteristics of the other flows. When sending at rates below the TCP-compatible rate, the unresponsive flow in our experiments underestimates the fair share since it does not aggressively probe for available bandwidth, in contrast to TCP and TFRC. Note that the effect is inherently coupled to the drop-tail policy employed in the routers and diminishes as the multiplexing level increases. Furthermore, if Active Queue Management (AQM) strategies such as RED are employed in the routers, the conditions become comparable to that of the artificial channel with a steady and independent loss process.

However, the aforementioned effect might severely impact the behavior of a receiver participating in a layered multicast transmission in case that gateways apply drop-tail queuing. When the receiver joins the base layer and only very few flows are competing for the same bottleneck, the TCP-fair level is very probably above the base layer rate. Hence the corresponding multicast receiver will overestimate the congestion level and behave very conservatively. While this does not negatively impacting TCP, the multicast receiver might never converge to a fair subscription level by joining appropriate multicast groups. As a consequence, we would recommend an operational equation-based multicast protocol to implement additional probing mechanisms to also address this issue.

5.7 Summary

In this chapter, we systematically investigated an equation-based approach for fair share estimation with respect to its applicability to multicast transmission. The approach grounds on an analytical model for the steady-state throughput of TCP and is based on round-trip time and loss rate estimation. Due to its salient performance in the unicast streaming domain it is considered a promising approach also for multicast transmission of continuous media. Since its performance particularly relies on the underlying techniques for loss measurement, we focused our study on the corresponding mechanism

The evaluation in this section has been performed by means of extensive network simulations. Therein, the conditions of the multicast scenario are modeled by unresponsive flows, which operate at transmission rates that are decoupled from receiver or network feedback. While the corresponding receivers have implemented the loss and rate estimator, the tight rate-control loop used in the unicast case is lacking. Hence, a flow's rate is not adjusted constantly to the actual network conditions, and its operating point might noticeably diverge from the TCP-compatible level. This abstracts quite well the multi-rate multicast scenario, where the limited number of corresponding group rates obviously cannot accommodate the conditions of highly heterogeneous receivers and data paths.

In a first set of experiments, the performance of the original algorithm for loss estimation has been intensively studied over a variety of simulation configurations. The results confirm that the original approach performs reasonably well over a wide range of parameter setting for open-loop transmission scenarios. However, its limitations become evident in high-multiplexing environments with moderate to high packet drop rates. End systems receiving data at rates that are above their fair share tend to overestimate the TCP-compatible rate. The overestimation effect is boosted as the transmission rate and packet drop rate increases. As a consequence, receivers participating in a multi-rate multicast session might report an overestimated TCP-compatible rate and choose a subscription level that is highly unfair.

After analyzing the origin of the observed effect, we devised modifications to the loss estimation procedure and introduced the concept of loss event impacts. The results of the subsequently performed set of experiments clearly demonstrated how our algorithm can outperform the original technique under the aforementioned operational regime. The modified algorithm fairly measures the level of congestion on an end-to-end path under a wide range of parameter settings. Thus, it may be employed by a receiver in a multi-rate multicast session (a) to provide the sender with a relatively accurate fair share estimate for the server to perform stream optimization (see Chapter 3); and (b) to choose the appropriate multicast subscription level based on the estimated TCP-compatible rate.

Nevertheless, our study also showed that the approach has limitations in environments of a very low degree of statistical multiplexing with gateways implementing drop-tail policy for queue management. For very low packet rates, our modification fails to improve the original algorithm and quite similarly tends to underestimate a receivers's fair share. In this particular case, a multicast receiver performing group subscription based on the fair share estimate might behave very conservatively. While this does not harm the cross traffic, it might lead to unfair situations for the multicast receiver. However, this effect is intrinsically coupled to the drop-tail policy applied in intermediate nodes and can be suppressed using AQM mechanisms such as RED.

It is important to emphasize that despite its limitations, the modification devised in this chapter can improve the original loss and fair share estimator while never performing worse than it.

Chapter 6

Subscription Level Management

In this chapter, we show how the TCP-compatible rate estimated with the equation-based model as discussed in Chapter 5 can be utilized for receiver-driven rate adaptation. In order to avoid frequent join and leave decisions in steady state while being responsive to changing conditions, a subscription level management strategy is developed.

Following the discussion of the shortcomings of a simple and straightforward strategy, a timer-based technique is derived in a first place, which is later extended to dynamic timers. The parameters of the proposed mechanism are tunable such that its responsiveness to congestion indications as well as its aggressiveness regarding the allocation of available network resources can be adjusted independently.

For the purpose of performance evaluation, we implemented our strategy and the related mechanism in a protocol framework by extending the functionality of the network simulator *ns-2*. The implementation also includes a module for estimating a receiver's fair share as discussed in Chapter 5. The resulting framework provides a powerful tool for studying the performance of our approach and to compare it with Receiver-Driven Layered Congestion Control (RLC), which is the most cited control scheme for layered multicast. Using the framework, we evaluate the behavior of both schemes in terms of smoothness in steady state for heterogeneous loss patterns and link delays, and test its reactiveness to transient congestion.

The roadmap of this chapter is shown in Figure 6.1.

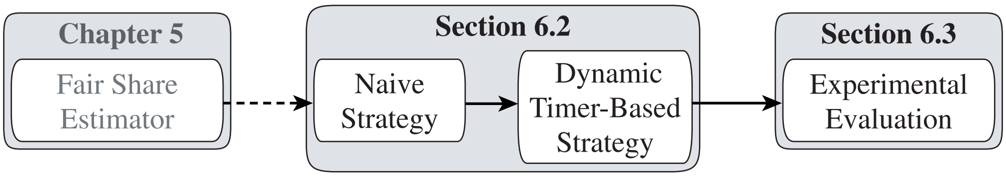


Figure 6.1: Roadmap of the chapter on *Subscription Level Management*.

6.1 Motivation

Native multicast transmission inherently differs from its unicast counterpart. In IP multicast packets are forwarded along a distribution tree rather than a single path. The tree is rooted at the data source and it extends to reach each receiver in the multicast group. Thus, a key feature of IP multicast service is the level of indirection provided by its host group abstraction. More specifically, while unicast packets are routed based on the receiver address included in the packet header, a multicast source generally does not need to have explicit knowledge about receivers and their addresses. Instead, the source pushes packets to one or more IP group addresses and receiver-driven group membership is utilized for implicit control of data distribution. That is, interested receivers have to signal their interest in joining groups to the network by means of IGMP messages.

Multi-rate transmission schemes are very powerful in accommodating heterogeneous multicast receiver sets (see Chapter 3). Moreover, they enable receiver-driven rate adaptation to the local capacity by receivers adjusting their level of subscription. This technique has been adopted by several researchers and became the de facto standard approach originally described by McCanne et al. in [MJV96]. However, the strategies and mechanisms utilized for subscription management are crucial for the performance and scalability of a protocol. Reacting to each loss drastically in a TCP-like manner and frequently changing the subscription level might cause severe rate and quality fluctuations to the application as well as increased overhead in network signaling and router state maintenance. On the other hand, protocol responsiveness is important for the stability and fairness of the network.

6.2 Subscription Strategy

In layered multicast transmission the sender stripes the data to several multicast groups in a hierarchical manner. The source predefines the order of the layers and receivers perform rate adaptation by subscribing and unsubscribing to the corresponding multicast groups. A subset of cumulative layers (i.e., multicast groups) is referred to as the subscription level. That is, a receiver subscribed to all layers 1 to l has subscription level l and hence it receives data at the corresponding group transmission rate g_l.

Since several receivers will have the same subscription level, they are members of the same multicast transmission group. The limited number of layers allows only for a relatively coarse-grained adaptation, which leads to a dilemma of fair competition, oscillations, and claim of fair share. When the fair share of a receiver is between two layers, the receiver may choose a subscription level that is either below or above the fair level. In such situations, long-term TCP-compatibility can be achieved by oscillating between the two subscription levels. However, this behavior should be avoided since streaming applications heavily profit from a smooth transmission. Furthermore, frequent changes of the subscription level impose a higher overhead in IGMP network signaling and router state maintenance.

Obviously, the smoothness and responsiveness of a layered multicast transmission scheme depends on the way receivers manage their subscription level. That is, they are heavily influenced by the way receivers join and leave multicast groups of a session in order to adapt their data rate and control congestion. We refer to this process as *subscription level management* and denote the underlying strategy as the subscription strategy.

The major focus of this section is the development of a subscription strategy. We start the discussion by introducing a straight forward *naive strategy*, extend it to a timer-based strategy, and finally present the derived *lazy strategy*.

6.2.1 Naive Strategy

A very simple strategy to make decisions for joining or leaving layers of a multicast session is to “naively” use the actual samples of the estimated fair share.

Assuming that a multicast receiver is subscribed to the lth level, it then belongs to the receiver transmission group l and its total reception rate corresponds to g_l. Furthermore, it is assumed that the current level l is less than the maximum possible subscription level L (i.e., number of layers) provided by the multicast session. As a result, when the estimated fair share χ of the receiver exhibits the transmission rate g_{l+1} of the next subscription level, the receiver will immediately trigger a group join message and change its subscription level. Similarly, as soon as χ falls below the current transmission rate g_l, the receiver will immediately trigger a leave action to change its subscription level to $l-1$.

This strategy is simple to implement and has been recently applied in a layered multicast protocol [LLZ02]. However, this strategy is very sensitive to fluctuations of the fair rate estimation caused by round-trip time and loss rate variations. Although smoothing is applied to both round-trip time and loss event rate estimation, the resulting estimates still might experience noticeable variations [RLSS03, RSS03, WBB04]. Consequently, when the average calculated fair share estimate $\bar{\chi}$ is slightly below or above the rate g_l of a transmission group, variations of χ might easily cause the samples to frequently exceed or fall below g_l even in steady state. This is schematically depicted in Figure 6.2 for the case of $\bar{\chi} < g_l$. Applying the naive subscription strategy in this case consequently might lead to oscillatory subscription behavior and increasing IGMP signaling overhead.

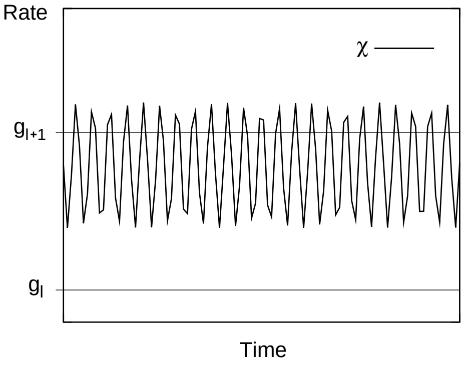


Figure 6.2: Schematic illustration of the variations of the estimated fair share χ of a receiver. The frequent crossing of the group transmission rate g_{l+1} will result in subscription level oscillations when applying the naive subscription strategy.

6.2.2 Join and Leave Timers

Smoothing the fair share estimate intuitively represents a straight-forward approach to overcome the aforementioned limitations of the naive subscription strategy. However, it can significantly decrease the responsiveness of the mechanism. Moreover, the values for the smoothing parameters of the original rate estimator have been thoroughly chosen as a result of extensive experiments (for a discussion see Chapter 5).

Therefore, the introduction of join and leave timers is considered for solving the problem. In this approach, the estimated TCP-compatible fair share χ exceeds the transmission group rate g_{l+1} of a receiver's next subscription level, a join timer is triggered instead of directly triggering a join action. If within a period τ_{join} the calculated rate χ keeps a value higher than g_{l+1}, then the timer triggers the receiver to subscribe to level $l+1$. Otherwise, if during this period χ falls below g_{l+1}, the timer is canceled and the receiver keeps its current subscription level. Similarly, when χ falls below the receiver's current transmission group rate g_l, a leave timer (τ_{leave}) is triggered instead of immediately sending an IGMP leave message.

Figure 6.3 schematically illustrates the basic concept of join and leave timers. At t_{join} the estimated fair share χ exceeds the group rate g_{l+1} of the next subscription level, which triggers a join timer. Since χ stays above g_{l+1} for a period of time longer than τ_{join}, the join timer triggers the receiver to subscribe to the next level. Similarly, at t_{leave} a leave timer is triggered that expires and triggers the receiver to subscribe to level l again. Subsequently, at t'_{join} the calculated fair share again triggers a join timer. However, this time χ undershoots g_{l+1} within a period shorter than τ_{join}. Hence, the join timer is canceled and the receiver keeps its current subscription level l.

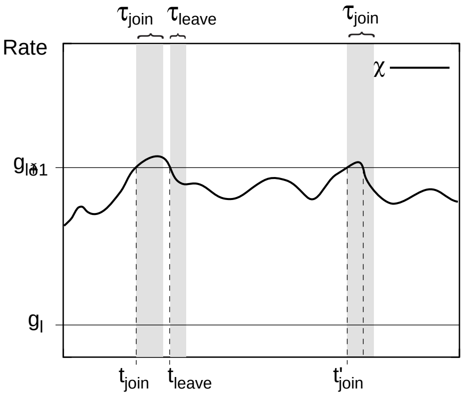
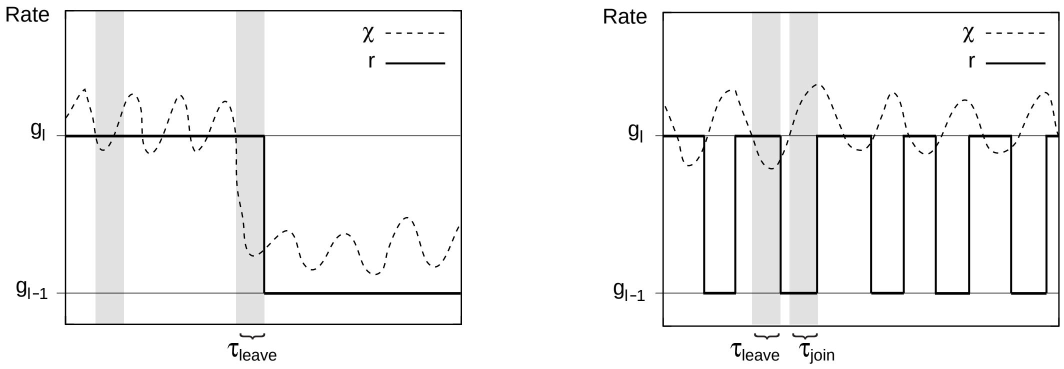


Figure 6.3: Schematic illustration of the concept of join and leave timers.

The timer-based strategy has two major benefits: (1) it can prevent from frequent subscription level oscillations, and (2) it allows for independently tuning the aggressiveness and responsiveness through the value of the join timer and the value of the leave timer, respectively.

However, appropriately configuring the timers to static values is a very difficult task because smoothness is generally traded for reactivity (i.e., responsiveness and aggressiveness). The periodicity and the pattern of the oscillations of the estimate fair share in steady state heavily depend on a priori unknown factors (e.g., loss pattern, round-trip time, multiplexing level, etc.; see Chapter 5).

To give an example, consider setting the join and leave timer to τ_{join} and τ_{leave}, respectively, such that the control scheme performs smoothly and responsively under certain environmental conditions. Under different conditions, however, it might become too sensitive and oscillate. This is demonstrated in Figure 6.4.



(a) smooth and reactive behavior

(b) oscillatory behavior

Figure 6.4: Operational rate and subscription behavior of a flow using the timer-based strategy for subscription level management in two different scenarios. The join and leave timers in both scenarios are constant.

6.2.3 Lazy Strategy

While the timer-based subscription level management presented in the preceding section is a promising approach, there is still an unsolved issue. The latter is attributed to the boolean operation for join and leave decisions over predetermined and static time periods. That is, the mechanism checks for $\chi > g_{l+1}$ and $\chi < g_l$, respectively, and does not consider how much the both values differ.

To improve the performance of the subscription level manager, the timer-based mechanism is extended by means of dynamic join and leave timers. The general idea is to allow for "lazy" decisions, as long as the difference exhibits a low value. That is, if the estimated fair share only slightly exceeds the group rate of the next subscription level or it is only slightly below the current operational rate, respectively, decisions are postponed.

Basically, the idea is to adjust the timer dynamically as a function of the actual distance of the estimated fair share and the group rate of the next subscription level $l+1$ for join attempts, respectively, the group rate of the current subscription level l for leave attempts. The higher the distance of both values, the faster a decision should be made.

Whenever a timer event occurs, the join and leave timers are computed according to the following formulas:

$$\tau_{join} = \hat{\tau}_{join} \cdot \lambda_{join}(\delta_{join}), \tag{6.1}$$

$$\tau_{leave} = \hat{\tau}_{leave} \cdot \lambda_{leave}(\delta_{leave}). \tag{6.2}$$

where $\hat{\tau}$ denotes the maximum value for the corresponding timer, λ the *laziness* factor, and δ the rate distance metric. Thereby, λ is a function of δ, and τ_{join} and τ_{leave} are constrained to the range $[0, \hat{\tau}_{join}]$ and $[0, \hat{\tau}_{leave}]$, respectively.

Furthermore, let us define the normalized rate distance functions δ_{join} and δ_{leave} to map from the estimated distance to a value in the interval $[0, 1]$:

$$\delta_{join} = \frac{\chi - g_{l+1}}{g_{l+1} - g_l} \quad \text{for} \quad g_{l+1} \leq \chi < g_{l+2}, \tag{6.3}$$

$$\delta_{leave} = \frac{g_{l-1} - \chi}{g_l - g_{l-1}} \quad \text{for} \quad g_{l-1} \geq \chi < g_l. \tag{6.4}$$

To configure the sensitivity of the subscription strategy to the computed rate distance, we introduce a *sensitivity function* ζ. For the latter, any wide-sense increasing function of δ can be chosen according to:

$$\zeta : [0, 1] \to [0, 1],\ \delta \mapsto \zeta(\delta),\ \frac{d}{d\delta}\zeta \geq 0. \tag{6.5}$$

The laziness λ is then computed as the compliment of the sensitivity function as follows:

$$\lambda(\delta) = 1 - \zeta(\delta). \tag{6.6}$$

From Equation 6.6 it follows that λ is bounded to values from the interval $[0, 1]$, so that $0 < \tau \leq \hat{\tau}$ as required.

Using the *lazy strategy* turns the formerly static parameters τ_{join} and τ_{leave} into dynamic values. It provides a flexible mechanism that allows for postponing join and leave decisions when the fair share estimate gives only a weak indication for choosing the next higher or lower subscription level. The laziness of the mechanism can be tuned by configuring $\hat{\tau}_{join}$ and $\hat{\tau}_{leave}$, and particularly by the provided sensitivity functions ζ_{join} and ζ_{leave}.

In preliminary experiments, we obtained promising results in terms of smoothness reactivity by heuristically choosing $\zeta_{join} = \delta_{join}$ and $\zeta_{leave} = \sqrt{\delta_{leave}}$. Figure 6.5 shows the resulting sensitivity and laziness as functions of the normalized rate distance.

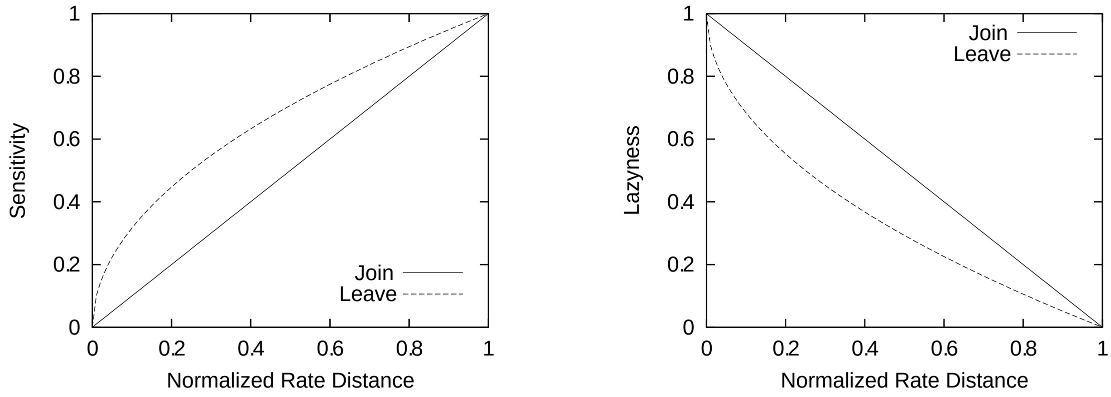


Figure 6.5: *Sensitivity* and *laziness* as functions of the normalized rate distance.

6.3 Experimental Evaluation

Having developed a novel subscription level management mechanism, the goal of this section is to prove the feasibility of the concept. Basically, we attempt to identify and highlight the strengths and weaknesses in terms of smoothness, responsiveness, and TCP-fairness of the lazy subscription mechanism in conjunction with our modified rate estimation algorithm (see Chapter 5). In order to compare our work with existing solutions, we confront it with RLC [VRC98], which currently is the most prominent layered multicast control scheme.

6.3.1 Overview

Following the methodology of Chapter 5, we employ network simulations using *ns-2* for the aforementioned evaluation. Therefore, we integrated the lazy group subscription mechanisms into our Adaptive Rate Control Framework (ARCF) (see Appendix C) assembling a multi-rate multicast control scheme named Equation-Based Layered Multicast (ELM). The latter makes evaluation of our mechanism tractable and comfortable. Furthermore, the availability of an implementation of RLC in *ns-2* allows for direct comparison of both schemes.

In order to efficiently reach our aforementioned evaluation goal and objectives, the following systematics applies to this section:

1. For analysis of performance in terms of smoothness, we use an artificial channel[1] for modeling steady-state conditions, similar to the experiments in Chapter 5. Recall that the fair share estimator calculates a rate that models the steady-state throughput of TCP. It is a function of the loss rate and the round-trip time. Thus, we investigate smoothness and TCP-fairness under different steady-state conditions, especially for (a) heterogeneous link loss rates and (b) heterogeneous round-trip times.

2. To analyze behavior in a contention environment, bandwidth-limited bottlenecks are used. If a small number of flows aggressively competes for available bandwidth (for example, using TCP-like AIMD probing schemes) dynamic loss patterns and fluctuations of the round-trip time are caused. As a consequence, the fair share estimator can exhibit a relatively pronounced degree of variations and oscillations. This represents a very challenging environment in terms of smoothness for an equation-based scheme even for contention situations with a constant number of flows. Furthermore, if suddenly more flows have to share the same resources, transient congestion occurs. In these situations, control mechanisms have to provide a sufficient degree of responsiveness in order to assure TCP-compatibility.

Figure 5.6 shows a graphical overview of our evaluation approach. Following the introduction of the applied fairness metric, we present the simulation configuration, and discuss the results of the simulations subsequently.

---

[1] Given properly provisioned link capacities and queue buffers, the artificial channel allows for configuring a packet drop rate that is constant and equal for all flows independent of their characteristics. It is commonly used for abstraction of an environment with a high level of statistical multiplexing.

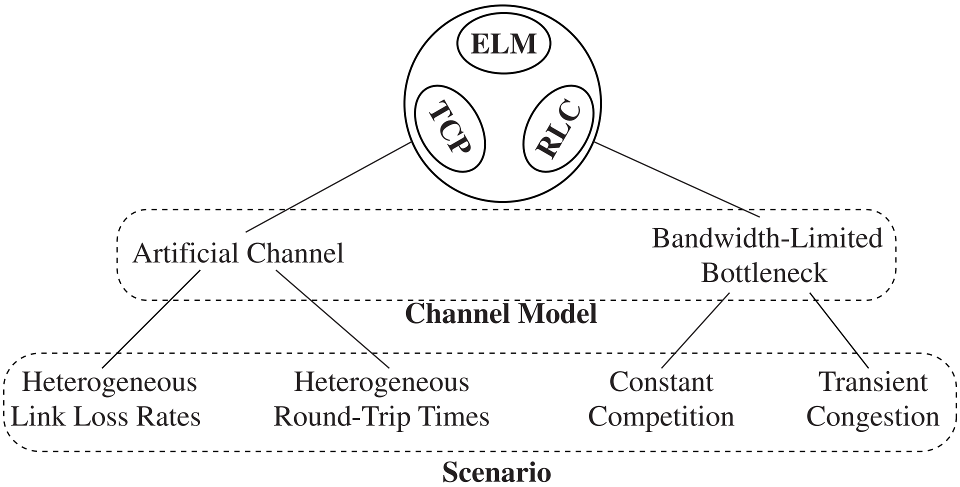


Figure 6.6: Overview of the experimental approach.

6.3.2 Metrics

In order to give quantitative statements about the performance of ELM and RLC, the following metrics are applied within our evaluation:

1. The smoothness and responsiveness of a scheme can be easily observed following the throughput progression over time. A straight-forward metric is the number of subscription level changes, that is, join and leave actions over time.

2. In contrast, the above metric provides insufficient means for accessing and comparing the fairness between flows of different sessions (inter-session) following the same path. Hence, our evaluation is based on the following ratio:

$$F_{flow} = \frac{\bar{T}_{flow}}{\bar{T}_{TCP}}, \tag{6.7}$$

   where $\bar{T}_{flow}$ denotes the average throughput of the investigated flow, and $\bar{T}_{TCP}$ the average throughput of a TCP flow running under equivalent conditions. This is a very commonly used metric for studying control schemes and "measuring" their TCP-fairness.

6.3.3 Simulation Configuration

For topology design and parameter configurations, our simulation setup follows the general recommendations for multicast simulations of members of the IETF Reliable Multicast Transport (RMT) working group [BFH$^+$00]. In particular, the tree topology of depth three depicted in Figure 6.7 has been chosen. It is comprised of the following nodes: a source node (S); intermediate gateway nodes (G1), (G2), (G3); and receiver nodes (R1), (R2), (R3).

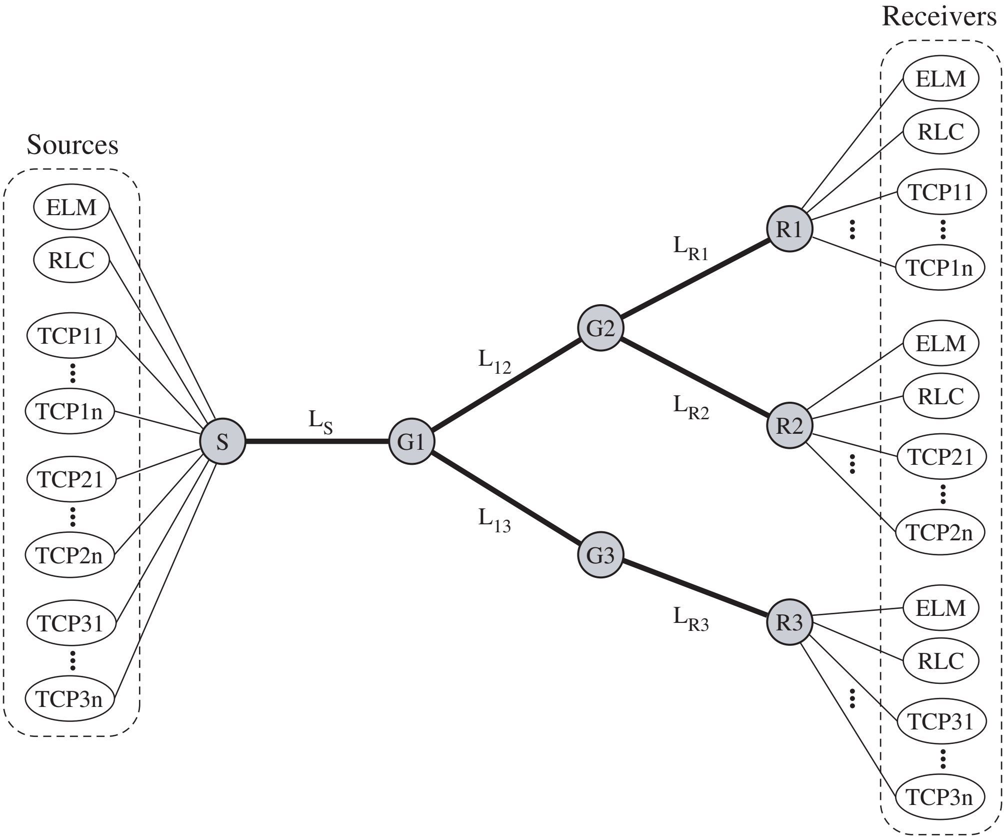


Figure 6.7: Simulation topology and setup.

One multicast sender agent of ELM and RLC is attached to the source node (S), and each receiver node (R1) to (R3) hosts an ELM and a RLC receiver agent. Furthermore, one TCP connection is set up between each source-receiver pair. For bandwidth-limited bottlenecks, additional TCP flows are configured for each source-receiver pair. The maximum values for the join and leave timers of ELM, $\hat{\tau}_{join}$ and

$\hat{\tau}_{leave}$, are heuristically set to 20 seconds. The RLC sender and receiver as described in [VRC98].

The default capacity of non-bottleneck links is set to 100 Mbps and their queue buffers are sufficiently provisioned in order not to effect flow behavior. The buffer size of bottleneck queues (drop-tail) is generally set to twice the bandwidth-delay product and link propagation delay to the default value of 10 ms. Depending on the simulation scenario, link properties are varied according to Table 6.1 and Table 6.2.

	Link Loss Rate or Capacity				
Simulation Scenario	L_{12}	L_{13}	L_{R1}	L_{R2}	L_{R3}
Heterogeneous Packet Drop Rates	1 %	3 %	1 %	-	-
Heterogeneous Round-Trip Times	1 %	1%	-	-	-
Constant Contention	-	-	3 Mbps	6 Mbps	12 Mbps
Transient Contention	-	-	3 Mbps	6 Mbps	12 Mbps

Table 6.1: Constraining link parameter values for the simulation scenarios. Default values are 100 Mbps and 0 %, respectively.

Simulation Scenario	L_S	L_{12}	L_{13}	L_{R1}	L_{R2}	L_{R3}
Het. Round-Trip Times	10 ms	10 ms	120 ms	105 ms	5 ms	120 ms

Table 6.2: Configuration for the scenario of heterogeneous round-trip times. Capacity of all links is set to the default values 100 Mbps.

All experiments have been conducted using a constant packet size of $1,000$ Bytes and repeated 15 times in order to provide statistical significance to the calculated mean. The required randomness is introduced by varying the starting time of the flows. Since exponentially-distributed layers are required for best performance of RLC, the base layer in all simulations is set to 128 kbps and each subsequent subscription level doubles the overall data rate. Hence, it follows for the subscription group rate $g_l = 2 \cdot g_{l-1}$.

Recall that the goal of this chapter is the investigation of a receiver-side subscription management strategy. The instances of ELM are not performing stream optimization as proposed in Chapter 3 in order to avoid interference with sender-side rate adaptation effects. Moreover, this allows for direct comparison with RLC, which is solely receiver-driven.

The results are discussed in the following subsections based on a single, representative simulation trace per scenario.

6.3.4 Artificial Channel

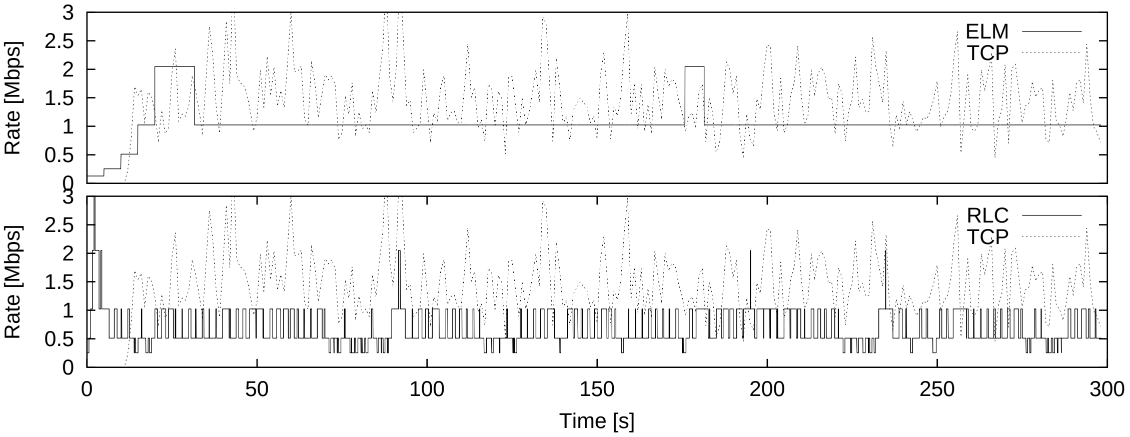
For investigating smoothness and TCP-fairness under different steady-state conditions for heterogeneous link loss rates and heterogeneous round-trip times, each receiver node hosts a single instance of ELM, RLC and TCP simultaneously, and the link properties are set according to Table 6.1 and Table 6.2 of Section 6.3.3.

Heterogeneous Loss

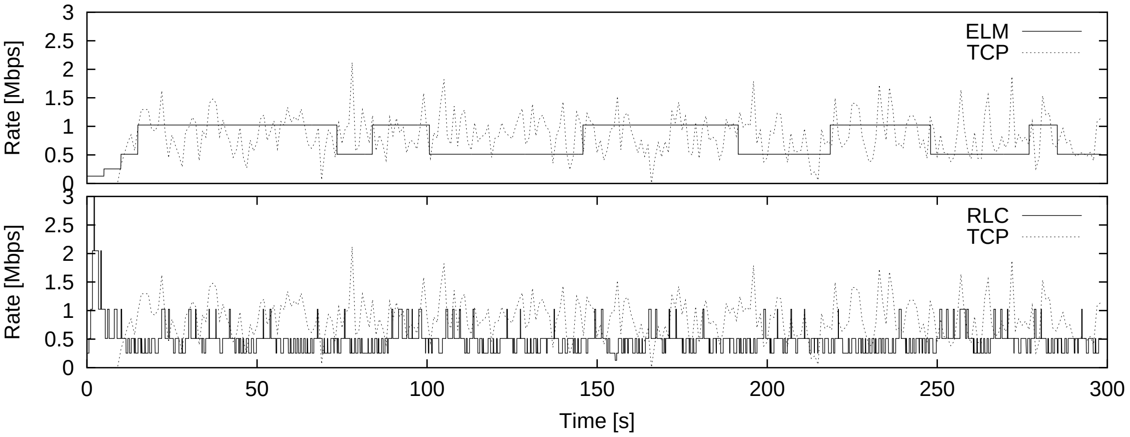
The subject of the first set of experiments is the protocol behavior under varying congestion levels. The latter is modeled by artificial channels with heterogeneous loss rates.

Figure 6.8 shows the traces of the ELM and RLC data rates (dotted lines) of the receivers at different receiver nodes. For comparison, the traces also include the throughput achieved by the TCP receiver located at the corresponding node. It is evident that ELM converges to a TCP-compatible subscription level and only very infrequently changes it. In general, the ELM receiver instance sticks to its actual subscription level and conducts very few subscription level changes. This leads to a stable and smooth operational behavior, while the achieved data rate and fairness in all three cases is reasonable high. In contrast, the RLC receiver reacts very sensitive to individual packet losses and frequently probes for available resources. Hence, it carries out significant more join and leave actions compared to ELM.

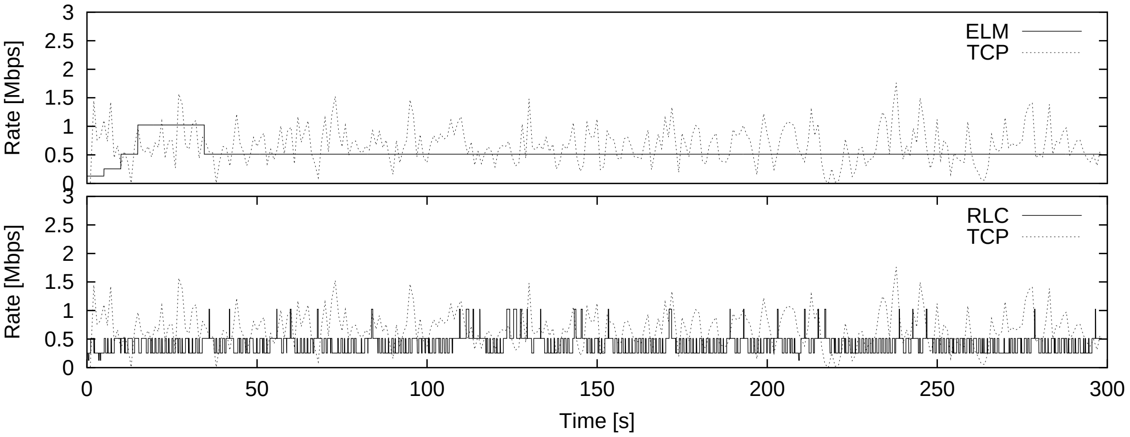
We have summarized the normalized throughput, the fairness index, and the ratio of ELM and RLC in terms of subscription actions in Table 6.3. Clearly, RLC performs also worse than ELM in terms of achieved data rate and fairness, although it employs a more aggressive probing scheme. Note that the fairness index of RLC appears to be slightly lower than the results reported by Vicisano et al. [VRC98] for drop-tail and RED queuing policies in a bandwidth-limited bottleneck scenario. The conditions on the artificial channel appear to be RLC-unfriendly, while ELM can deal with them very well due to its TCP-model based estimator and the employed subscription strategy.



(a) receivers at node R2 ($p_{drop} = 1\,\%$)



(b) receivers at node R1 ($p_{drop} = 2\,\%$)



(c) receivers at node R3 ($p_{drop} = 3\,\%$)

Figure 6.8: Comparison of ELM and RLC for different packet drop probabilities.

	R1	R2	R3
$\bar{T}_{TCP}$	912 kbps	1,429 kbps	668 kbps
F_{ELM}	0.859	0.753	0.760
F_{RLC}	0.515	0.449	0.599
subscriptions ELM vs. RLC	1:35	1:30	1:71

Table 6.3: Performance comparison of ELM and RLC under heterogeneous packet drop rates. Values are averaged over a simulation time of 300 s.

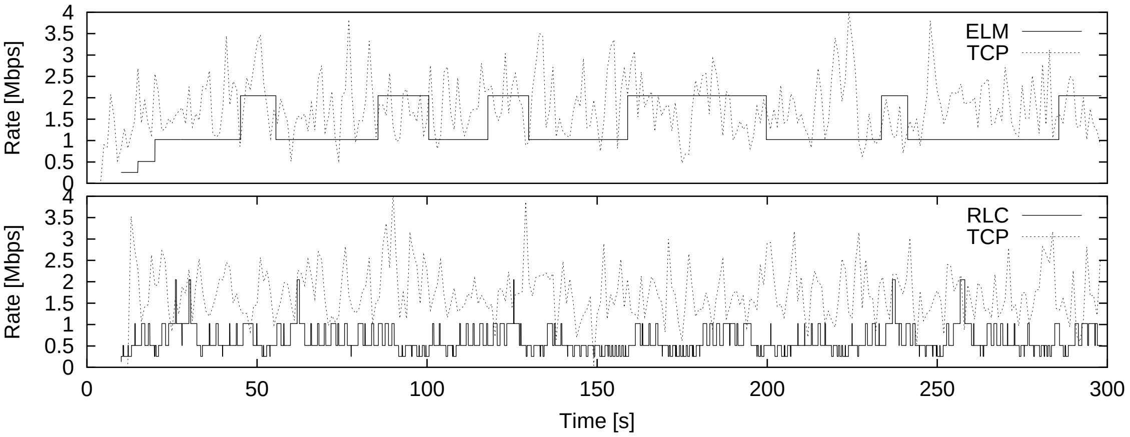
Heterogeneous Round-Trip Times

In the following simulation setup heterogeneous round-trip times are configured setting link propagation delays according to Table 6.2. As a result, receivers at node (R1) through (R3) experience a round-trip time of 250 ms, 50 ms, and 500 ms, respectively. Hence, our simulation setup covers a wide range of round-trip times. The packet loss rate, on the other hand, has been set to a constant and equal value of 1 % for all receivers in order to allow for isolated examination of the effects. The traces of one representative simulation run are depicted in Figure 6.9.

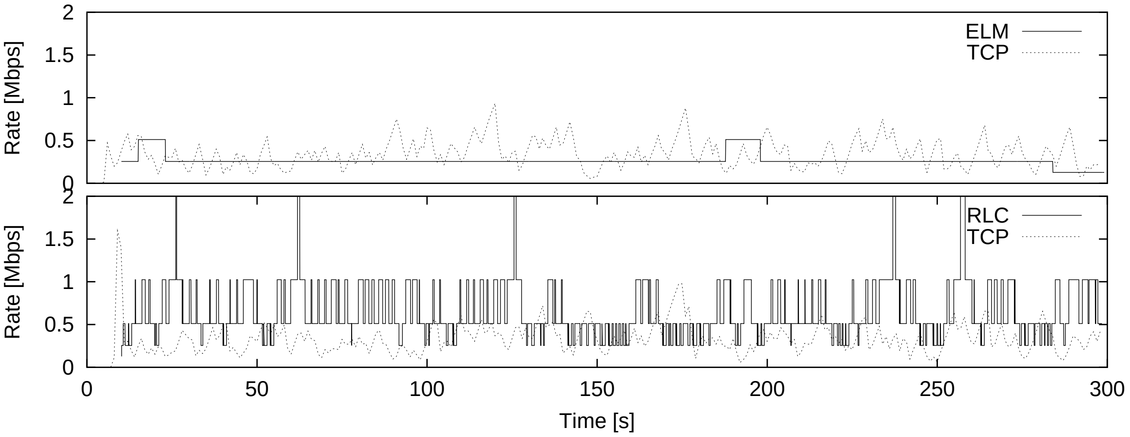
Figure 6.9(a) shows how lower round-trip times pose a problem to RLC for achieving a TCP-fair average rate. The average throughput of TCP is 1,726 kbps and RLC on average only reaches a value of 652 kbps. Meanwhile, ELM converges to the 4th subscription group at 1,024 kbps and reaches an average data rate of 1,417 kbps. Although the TCP-fair level is close to the next higher subscription level, ELM exhibits oscillations between subscription level 4 and 5 only at a low frequency. The resulting ratio for join and leave decisions of ELM and RLC is around 1:21 on average.

As Figure 6.9(b) and Figure 6.9(c) show, ELM keeps its smooth and fair behavior also for higher values of the round-trip time. On the other hand, RLC exhibits oscillatory behavior and reaches an average throughput that is substantially higher than that of the TCP flow running under the same conditions. The quantitative results of our experiments are summarized in Table 6.3.

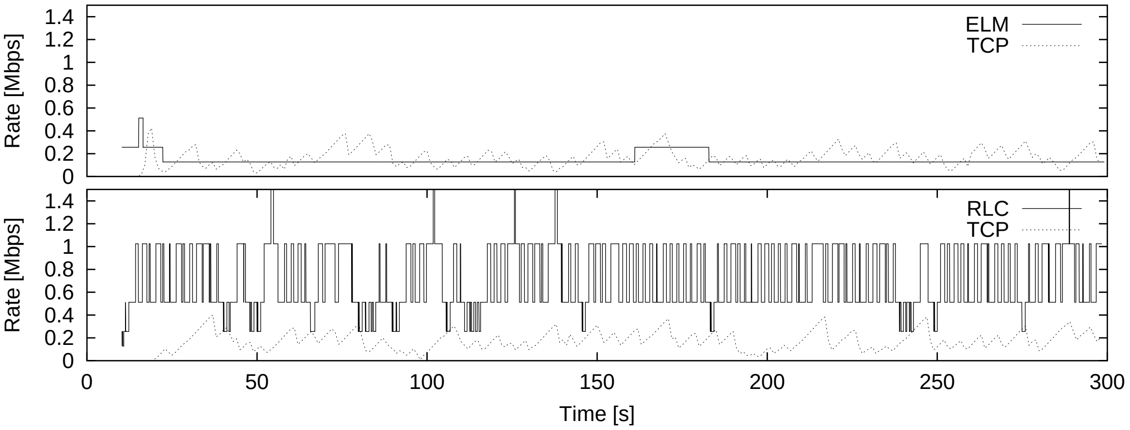
For an interpretation of the results, recall that the TCP throughput is inversely proportional to the round-trip time. This behavior is also reflected in the TCP model used for fair share estimation in ELM. Consequently, ELM implements a mechanism for estimating the round-trip time. RLC on the other hand is only based on loss detection and uses an probing mechanism based on static timers ignoring



(a) receivers at node R2 ($r_{RTT} = 50\,\text{ms}$)



(b) receivers at node R1 ($r_{RTT} = 250\,\text{ms}$)



(c) receivers at node R3 ($r_{RTT} = 500\,\text{ms}$)

Figure 6.9: Comparison of ELM and RLC for different round-trip times.

	R1	R2	R3
$\bar{T}_{TCP}$	342 kbps	1,726 kbps	176 kbps
F_{ELM}	0.763	0.821	0.750
F_{RLC}	1.843	0.378	3.559

Table 6.4: Performance comparison of ELM and RLC under heterogeneous round-trip times. Values are averaged over a simulation time of 300 s.

the actual round-trip time. Hence, our results confirm the reported pathological behavior of RLC [LB00a] in terms of TCP-compatibility.

6.3.5 Bandwidth-Limited Bottleneck

The simulations based on artificial channel conditions explored the behavior of ELM and RLC under conditions expected in environments with a high level of statistical multiplexing. The objective of the following section is to shed light on the performance of the scheme when they directly compete for available resource with a small number of greedy[2] TCP flows on bandwidth-limited bottleneck.

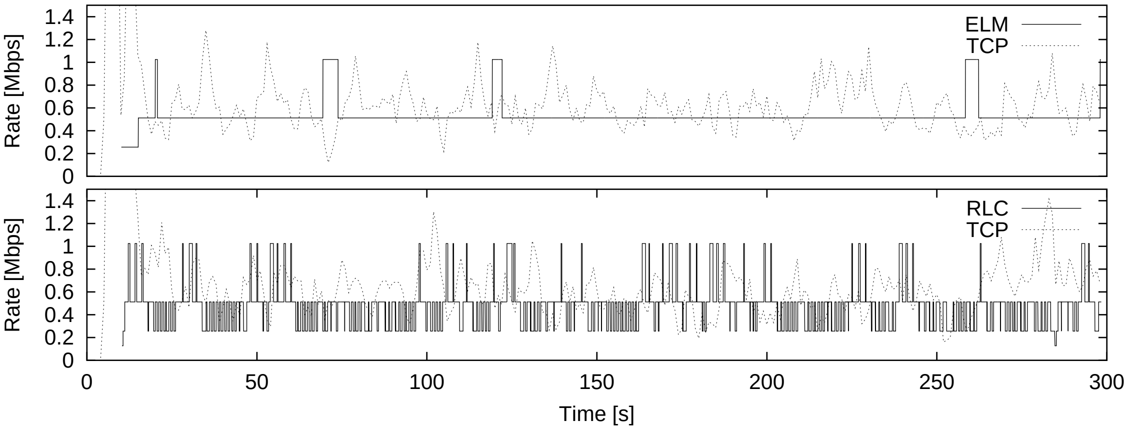
In the following experiments, link propagation delays are set to the default value so that every receiver node experiences a round-trip time of approximately 60 ms. The bottleneck link capacities are configured according to Table 6.1. Each receiver node hosts one ELM agent, one RLC agent, and four TCP agents. Since the particular objective of our investigation is the performance of ELM and RLC when competing with TCP, simulations for both are performed independently. That is, RLC is inactive for the experiments focusing on ELM, and vice versa.

Constant Competition

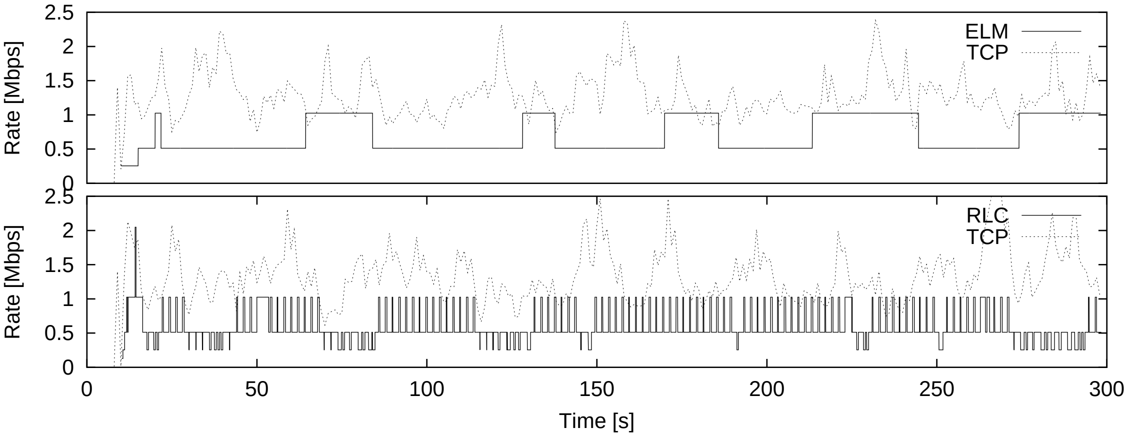
Figure 6.10 depicts the results obtained for a simulation in which the number of competing flows is constant. Particularly interesting is the observed smoothness of ELM despite the experienced degree of loss rate fluctuations. For example, once the ELM receiver attached to (R2) converged to its operational point, on average it performed a subscription action every 29 seconds, while RLC performed more than 20 times more join and leave decisions in the same time.

---

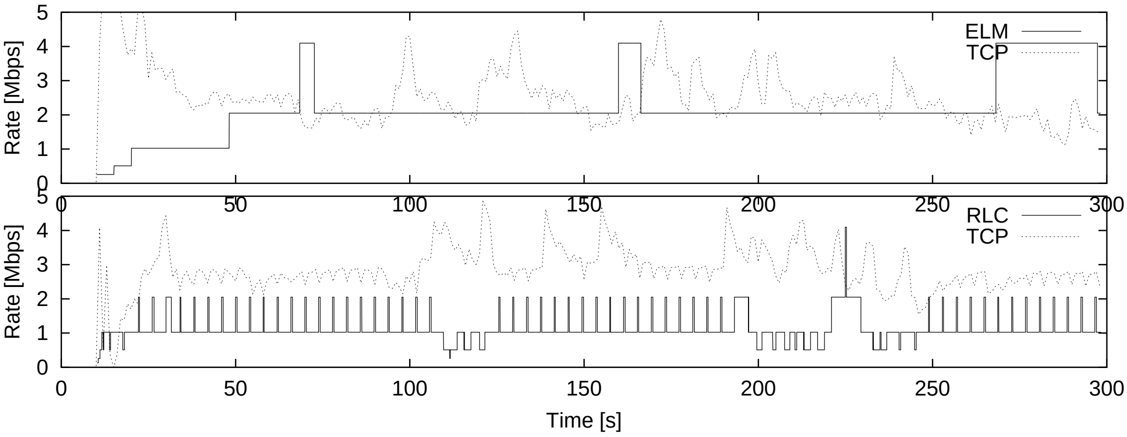
[2] If the transmission rate of a TCP flow is not limited by the application and it has sufficient data to send—for example, transfer of a very large file—, it will constantly probe for available bandwidth normally utilizing AIMD. Such a flow is usually referred to as being *greedy*.



(a) receivers at node R1 ($C = 3\,\text{Mbps}$)



(b) receivers at node R2 ($C = 6\,\text{Mbps}$)



(c) receivers at node R3 ($C = 2.4\,\text{Mbps}$)

Figure 6.10: Comparison of ELM and RLC for different round-trip times.

Comparing ELM's performance on the 3 Mbps and 6 Mbps bottleneck links yields to interesting results. The TCP-fair share in the former case is around 0.6 Mbps and slightly above the data rate of subscription level 3. Having an average throughput of 550 kbps, ELM achieves a very high level of TCP-fairness. For the other case, the TCP-fair share is around 1.2 Mbps and above the data rate of subscription level 4. However, the oscillations of the estimated fair share are quite pronounced causing early canceling of ELM's join timer and preventing it from joining to layer 4. Hence, it hardly attempts more aggressive to join layer 4 and achieves an average throughput of only 790 kbps.

Note that RLC for increasing bottleneck bandwidth capacities exhibits a lower degree of oscillations compared to the preceding results. This can be attributed to the larger queue buffers as follows. TCP flows use AIMD and increase their data rate at one packet per round-trip time, which takes more time to overflow the queue buffer. Consequently, packet drops occur less frequently and since RLC reacts solely on individual packet drops, it performs more smoothly.

Table 6.5 gives an overview of the results obtained for the discussed scenario.

	R1	R2	R3
$\bar{T}_{TCP}$	632 kbps	1,309 kbps	2,518 kbps
F_{ELM}	0.810	0.604	0.886
$\bar{T}_{TCP}$	645 kbps	1,353 kbps	2,768 kbps
F_{RLC}	0.741	0.441	0.402

Table 6.5: Performance comparison of ELM and RLC under constant contention with 4 TCP flows. Values are averaged over a simulation time of 300 s.

Transient Congestion

So far we have investigated only steady-state scenarios, where the number of competing flows is constant over time. In order to study the performance of ELM in terms of responsiveness to transient congestion, we introduce a disturbance to the system. That is, at a certain point in time the number of competing TCP flows is doubled.

Links are configured identically to the preceding simulations, so that (R1) through (R3) are behind a 3 Mbps, a 6 Mbps, and a 12 Mbps bottleneck. The start time t_{start} and the stop time t_{stop} of the additional cross-traffic are provided in Table 6.6.

Node	Start Time	Stop Time
R1	100	140
R2	200	220
R3	230	240

Table 6.6: Start and stop times of the additional cross traffic.

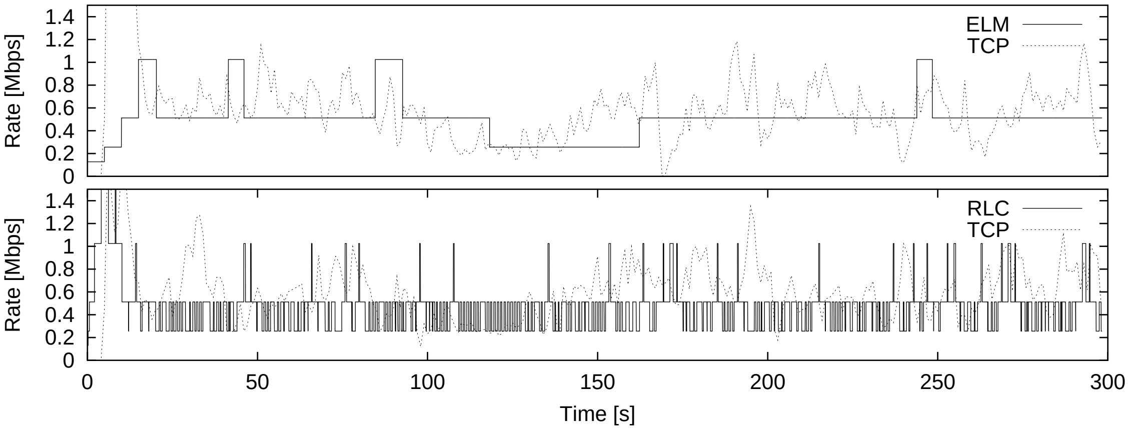
Figure 6.11 shows the traces of one experiment for ELM and RLC. Generally, we observe that while ELM performs very smoothly, it reacts with delay to transient congestion. For example, the theoretical fair share at the receiver node (R1) reduces from 600 kbps to 333 kbps during the congestion phase. In order to behave in a TCP-compatible manner, ELM is expected to react and leave the 3rd layer. As Figure 6.11(a) demonstrates, it might take more than 10 seconds before ELM triggers the necessary unsubscription. For higher data rates the performance in terms of responsiveness increases while aggressiveness decreases.

We emphasize that in all our simulation results ELM reacts to transient congestion by reducing its subscription to a fair level. Despite its delayed reaction, ELM does not starve the TCP-based cross-traffic and yields to a high degree of fairness. This indicates that the concept underlying ELM allows it to perform reasonably even under challenging conditions. However, further investigations for optimizing the sensitivity function and timer parameters are recommended.

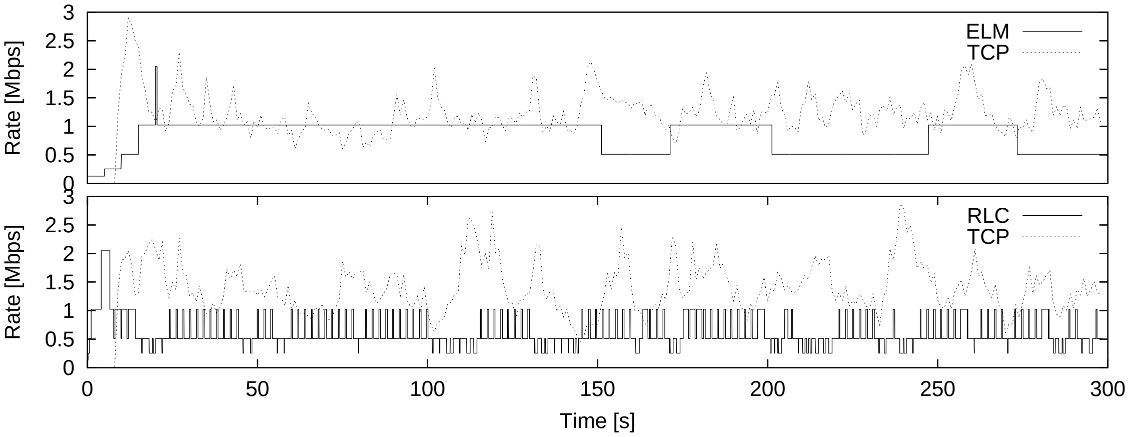
6.4 Summary

In this chapter, we developed a concept for receiver-driven subscription level management in multi-rate multicast control. Given the TCP modeling algorithm presented in Chapter 5, our timer-based strategy utilizes the calculated fair share estimate to decide on group join and leave action. Since the fair share estimator is intrinsically sensitive to fluctuations of the round-trip time and loss rate variations, dynamic timers are introduced in order to avoid rate oscillations in steady state.

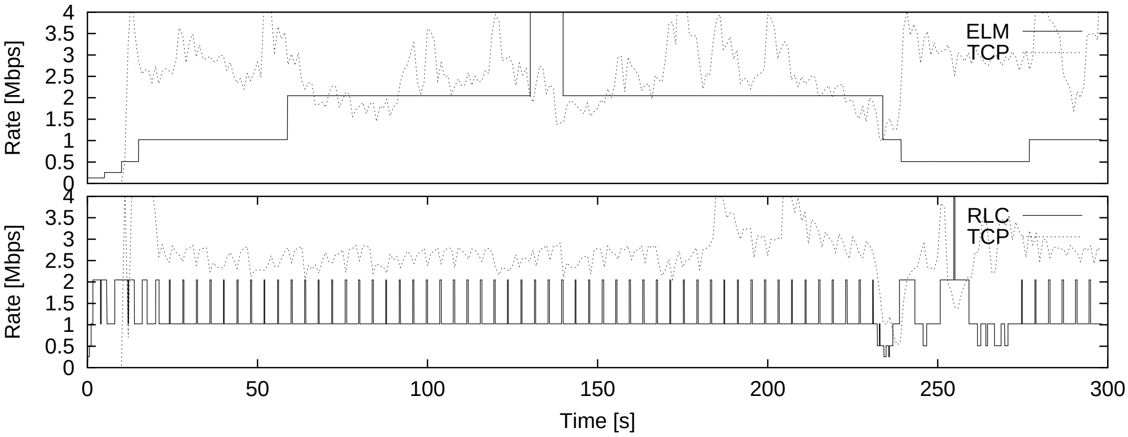
We have studied the feasibility of our approach by means of network simulations. The results obtained in our experiments indicate that our concept is very well suited to avoid subscription level oscillations under a broad range of network conditions. Consequently, quality fluctuations can be significantly decreased if the proposed subscription strategy is applied to video streaming applications. This finally results



(a) receivers at node R1 ($t_{start} = 100\,\mathrm{s}$, $t_{stop} = 140\,\mathrm{s}$)



(b) receivers at node R2 ($t_{start} = 200\,\mathrm{s}$, $t_{stop} = 220\,\mathrm{s}$)



(c) receivers at node R3 ($t_{start} = 230\,\mathrm{s}$, $t_{stop} = 240\,\mathrm{s}$)

Figure 6.11: Comparison of ELM and RLC for transient congestion.

in a higher degree of user satisfaction. Furthermore, the overhead in signaling and router state maintenance is reduced since frequent group management messages to the IP multicast network nodes are avoided.

In addition to its outstanding smoothness properties, our mechanism yielded a significant higher degree of TCP-fairness when compared to an existing and frequently cited control scheme. Moreover, it proved to be responsive to transient congestion even when reacting to changing conditions with some delay. To further improve the parameter setting, experiments are necessary that focus on the optimization of the underlying sensitivity function and the timer parameters, which is not within the scope of the thesis.

Chapter 7

Conclusion and Outlook

Facing the problems that surround multicast streaming of continuous media over the Internet, we started with our quest for mechanisms that can improve the current state-of-the-art. This chapter concludes our work and presents directions for future research.

7.1 Conclusion

Heterogeneity and scalability have been commonly named as challenging issues for multicast transmission in general. Further issues originate from the demand for TCP-compatibility to ensure stability of the Internet and the requirements of continuous media for smooth transmission. We tackled these issues within our work with a hybrid sender-based and receiver-driven approach for rate adaptation utilizing scalable-encoded video and layered multicast transmission.

We started with an assumption of global knowledge about the receiver bandwidth capabilities at the server. For the data source to optimally adjust the transmission rates, existing solutions rely on network-centric metrics. These metrics assume a linear correlation of the user satisfaction and the data rate, however, research in video quality measurement has indicated a non-linear relationship between both measures. Hence, we extended the state-of-the-art inter-session fairness metric to incorporate application aspects in terms of user utility and video quality. Moreover, we developed an appropriate optimization algorithm that has reasonable low complexity to allow for frequent computation. An experimental comparison of the optimization-based and adaptive allocation scheme with traditional, static strategies revealed that for heterogeneous receiver sets the former approach can significantly improve the overall satisfaction of the receivers.

Our optimization scheme for sender-side rate adaptation requires knowledge about the distribution of receiver bandwidth capabilities. To acquire this knowledge, the server needs to solicit feedback from the set of active receivers. Since feedback is a major source of the scalability issue, we developed a feedback control scheme based on probabilistic sampling. The sender then applies the optimization algorithm to the representative sample of receiver bandwidth capabilities. Our solution adopts an existing approach for estimating the session size and extends it with a dynamic algorithm for updating the estimate and the feedback probability. The presented mechanism is highly flexible and allows for controlling the amount of feedback traffic to a predefined rate within statistical bounds.

To reduce the inaccuracy introduced with sampling, we studied the applicability of statistical models for the bandwidth distributions measured on the Internet. With this approach, the parameters of the statistical model are derived from the feedback sample and optimization is performed according to the resulting approximation. Out of appropriate approximation functions, we found the Weibull function to provide the best results. The major conclusion that can be drawn from our experimental results is that the inaccuracy introduced with the feedback sampling approach is bounded to a very low value for different capability distributions. Furthermore, the Weibull-based approximation can improve the adaptation results for very small sample sizes.

The objective of the sender-side adaptation process is to increase a global intra-session metric and it is normally performed within periods of at least a few seconds to avoid oscillations. For short-term control of the data rate in order to react to congestion indication, we employed receiver-driven rate adaptation. However, the decisions of both mechanisms are based on the same algorithm for estimating the bandwidth capabilities of a receiver. For this purpose, we adopted the state-of-the-art algorithm for equation-based TCP-fair share estimation. In extensive simulations we showed that naively adopting the algorithm from closed-loop unicast to open-loop multicast transmission results in unfair behavior. Although other researchers reported pathological behavior under similar conditions, our work is the first that thoroughly investigated this phenomenon. Those new insights led to the proposal of an improved version of the algorithm, which reasonably well estimates the fair share over a wide range of conditions. Nevertheless, we also showed that there are inherent limitations to the equation-based approach, specifically in the case when a multicast flow competes with very few aggressive flows on a common bottleneck link.

Equation-based fair share estimation provides a very good platform for smoothly controlling the data rate of streaming video. Nevertheless, naively performing group subscription based on the actual estimate of the fair rate can easily yield oscillatory behavior. We tackled this problem by developing a novel strategy for subscription level management. Our approach introduces dynamic join and leave timers that can be tuned to adjust the smoothness and responsiveness of the adaptation mechanisms, respectively. The fair share estimator as well as the subscription mechanism have been implemented in a network simulator environment providing a framework for experimental investigation. Our simulation study shows that our approach outperforms one of the most frequently cited congestion control schemes in terms of smoothness and TCP-fairness.

The major conclusion we can draw from our research is that the combination of equation-based fair share estimation with hybrid multi-rate adaptation schemes is a promising approach for multicast streaming. We believe that our mechanisms as well as our findings lay foundation for designing comprehensive multicast rate adaptation solutions for continuous media streaming over the Internet.

7.2 Outlook

Our contributions advance the state-of-the-art in rate-adaptive multicast streaming and pave the way for further research towards scalable and stable multicast transmission schemes.

We presented a feedback scheme based on probabilistic sampling that can efficiently limit the feedback traffic within statistical bounds. In our scheme, each feedback round is independent and feedback values from former rounds are discarded at its beginning. Research in Internet measurement indicates that the conditions of a network path vary only slightly in a short and medium term. Hence, keeping history of feedback values and biasing feedback probability in favor of receiver's with sudden changes of congestion state is a possible direction of future work.

In the area of receiver-driven mechanisms for fair share estimation and rate adaptation, our contributions and findings provide solid ground for further investigation. We showed the limitations of the equation-based approach in the case of a very low multiplexing level on a bottleneck link. While our mechanism is conservative and does not harm competing TCP traffic, in the aforementioned situations it experiences a certain degree of unfairness. Future research might address this issue and further improve the performance of our mechanism.

The importance of providing point-to-multipoint delivery services has been recognized to be a major issue also for 3G telecommunications networks. Multicast audio and video streaming are envisioned as a major part of the service portfolio. This is reflected in recent efforts within the 3G standardization bodies, such as the Third Generation Partnership Project (3GPP), which led to the development of a 3G multicast architecture. The architecture is still in the process of standardization and it currently only supports a very simple service model. Since heterogeneity of network conditions can be identified as a common problem inherent to both the Internet and 3G networks, our work can be extended to provide 3G multi-rate multicast streaming services.

Appendix A

FGS Video Sequence Parameters

For the quantitative evaluation we resort to video test sequences from [DL03] and [DLR03]. The corresponding coding parameters and parameters for the square-root rate-distortion model are presented in the following table:

Video Sequence	Coding Resolution	Base Layer [kbps]	Model Parameters ν_1 [dB/kbps]	ν_2 [dB/$\sqrt{\text{kbps}}$]	ν_3 [dB]
Foreman	CIF (10 fps)	128	$2.50 \cdot 10^{-3}$	0.1423	29.15
Coastguard	CIF (10 fps)	128	$2.85 \cdot 10^{-3}$	0.1139	26.76
Earphone	CIF (10 fps)	128	$1.32 \cdot 10^{-2}$	-0.0910	33.02

Table A.1: Test sequences and parameters.

Throughout this thesis, we conduct all test assuming a maximum rate of $r_{max} = 2,560$kbps.

Appendix B

Bandwidth Capability Distributions

For the simulations conducted in Chapter 3 and Chapter 4 we generated receiver set populations and samples from different distributions. This allows us to evaluate the performance of our mechanisms under heterogeneous and changing conditions. The set of distributions we consider are described in the following.

B.1 Uniform Distribution

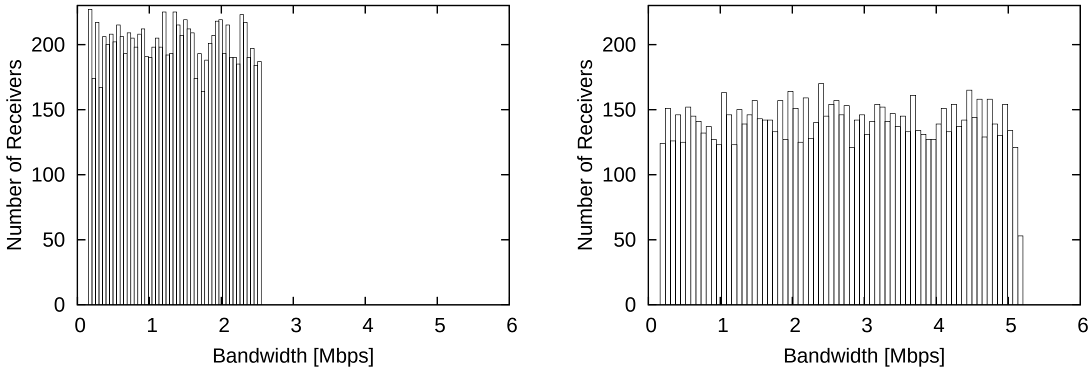
The uniform distribution $U(a;b)$ is commonly used if a random variable is bounded and no further information is available. Uniformly distributed receiver capabilities are distributed evenly over a defined range $[a,b]$ kbps. The values are generated according to the probability density function:

$$pdf(x) = \frac{1}{b-a}.$$

Since the values for the base rate r_{base} and the maximum rate r_{rmax} expressed in kbps span a rather large range, we resort to the discrete version of the uniform distribution. As a result receiver capabilities are following the probability mass function:

$$pmf(x) = \frac{1}{b-a+1}.$$

Figure B.1 illustrates two receiver sets that are uniformly distributed according to $U(128;2,560)$ respectively $U(128;5,120)$.



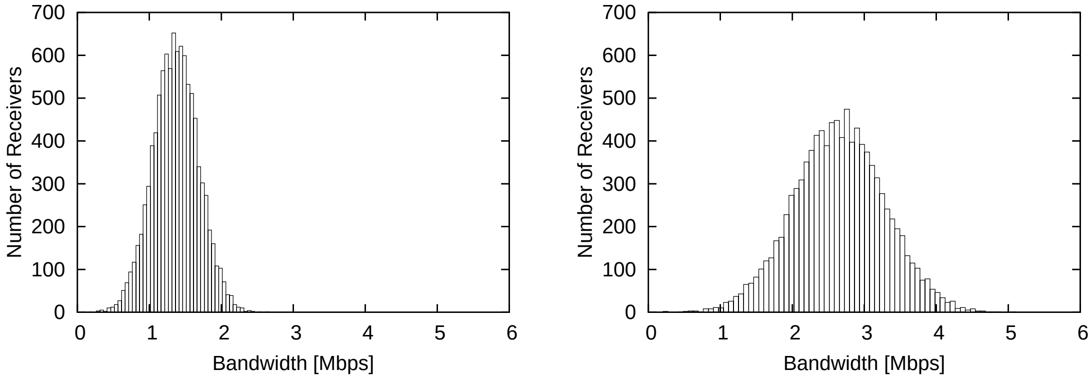
(a) range 128 . . . 2, 560 kbps (b) range 128 . . . 5, 120 kbps

Figure B.1: Histogram of receiver populations of size $N = 10,000$ following a uniform distribution.

B.2 Normal Distribution

Also known as the Gaussian distribution, the normal distribution is usually denoted with $N(\mu; \sigma)$. It is a symmetric distribution with mean value μ and standard deviation σ. A sample of n observations is generated according to the probability density function:

$$pdf(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}.$$



(a) $\mu = 1,344$ kbps and $\sigma = 320$ kbps (b) $\mu = 2,624$ kbps and $\sigma = 640$ kbps

Figure B.2: Histogram of receiver populations of size $N = 10,000$ following a normal distribution.

Figure B.2 illustrates the histogram of two receiver sets that are normally distributed according to $N(1,344;320)$ respectively $N(2,624;640)$, where μ and σ are measured in kbps.

B.3 Multi-modal Distribution

To model distribution with multiple distinct modes we resort to modeling each mode as being beta distributed $B(a;b)$. The shape parameters a and b are limited to positive real numbers. The beta distribution is used to represent random variables that are bounded within the range $[x_{min}, x_{max}]$. The probability density function of a beta distribution is defined as:

$$pdf(x) = \frac{\left(\frac{x-x_{min}}{x_{max}-x_{min}}\right)^{a-1}\left(1-\frac{x-x_{min}}{x_{max}-x_{min}}\right)^{b-1}}{\beta(a,b)} \qquad \text{for } x_{min} \leq x \leq x_{max},$$

with the beta function:

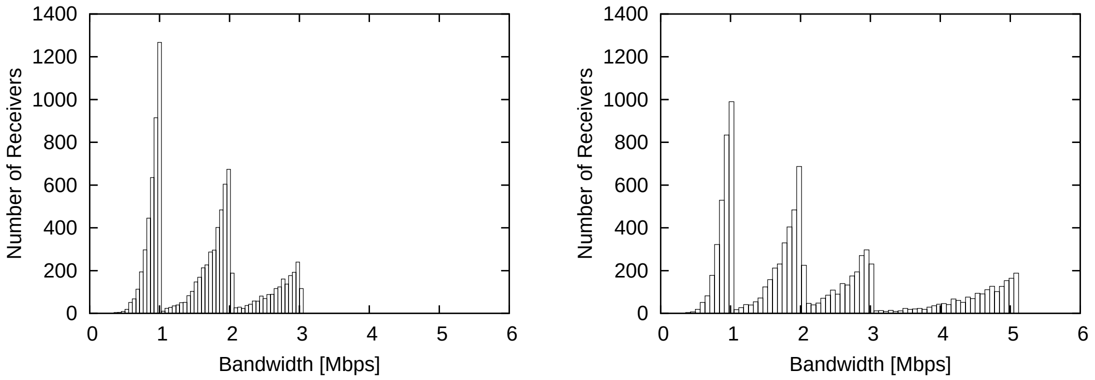
$$\beta(a;b) = \frac{(a-1)!(b-1)!}{(a+b-1)!}.$$

For a multi-modal distributed set we partition the receivers into m subsets. Thereby m denotes the number of modes and the fraction of receivers belonging to a subset is configured using the weights $w_1, \ldots, w_m$. The resulting multi-modal distribution can then be expressed by the weighted sum of m beta distributions:

$$\frac{\sum_i^m w_i B(a_i;b_i)}{\sum_i^m w_i}.$$

We choose the modes such that they reflect actual access technologies, particularly the three prevalent variants of Digital Subscriber Line (DSL): 1 Mbps, 2 Mbps, and 3 Mbps. Furthermore, by assigning higher weights to the lower bandwidth modes we assume that new (faster) technologies are more expensive and less popular than technologies that already established on the market for a longer time.

Figure B.3 illustrates two receiver sets that are multi-modal distributed according to the above assumptions. For the larger interval (Figure B.3(b)) we introduced an additional at the maximum transmission rate.



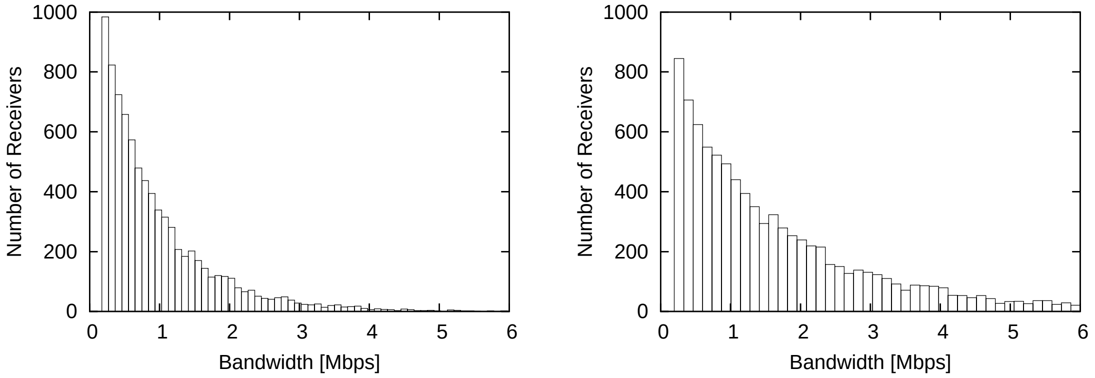
(a) range $128\ldots3{,}000$ kbps, weights $=\{2,2,1\}$ (b) range $128\ldots5{,}120$ kbps, weights $=\{3,3,2,2\}$

Figure B.3: Histogram of receiver populations of size $N = 10{,}000$ following a multi-modal distribution.

B.4 Weibull Distribution

The Weibull distribution is versatile distribution that is defined by an scale parameter a and an shape parameter b. The probability density function of the Weibull distribution is defined as:

$$pdf(x) = \frac{bx^{b-1}}{a^b} e^{-\left(\frac{x}{a}\right)^b}.$$



(a) $a = 738$ (b) $a = 1{,}476$

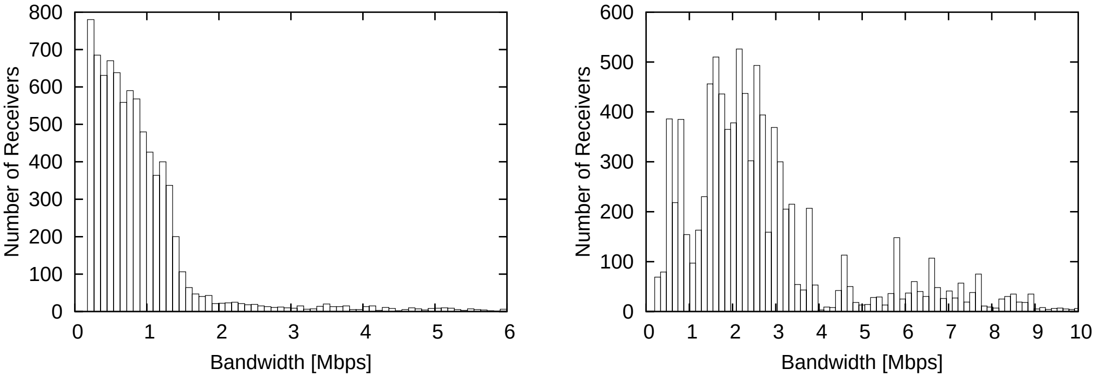
Figure B.4: Histogram of receiver populations of size $N = 10{,}000$ following a Weibull distribution.

The Weibull distribution can take on the characteristics of other types of distributions, based on the value of b: if $b = 1$ it equals an exponential distribution, and it is close to a normal if $b = 3.602$; for $b > 3.602$ and $b < 3.602$ it has a long right tail respectively a long left tail; it is L shaped for $b \leq 1$ and bell shaped otherwise. We use Weibull distributions to model measurement-based populations.

Figure B.4 illustrates the histogram of two receiver sets that are Weibull distributed with $b = 0.19$ and $a = 738$ respectively $a = 1,476$.

B.5 Measurement-Based Distribution

While so far we have derived distributions based on theoretical models, we also available bandwidth distributions as measured on by Paxson [Pax97c] and the PingER (Ping End-to-end Reporting) project [MC00]. We derived the non-continuous cumulative density functions for both empirical distributions, and generated receiver populations according to these. Figure B.5(a) and Figure B.5(a) show the histogram for populations of $N = 10,000$ receivers.



(a) Paxson

(b) PingER

Figure B.5: Histogram of receiver populations of size $N = 10,000$ following the distributions as derived from Internet measurements reported by Paxson [Pax97c] respectively the PingER-Project [MC00].

In the following, we provide a brief overview of the background information of the work of Paxson and the PingER project and the corresponding data.

B.5.1 Paxson Measurement

The results presented by Paxson in [Pax97b] are measured between 35 sites from 9 different countries. The sites include a diverse mixture of educational institutions, net work service providers and commercial companies. Although the measurements of this study were conducted almost a decade ago, they results are still a valuable source regarding bandwidth distribution characterization. While Paxson describes different aspects of Internet packet dynamics, we focus on the results regarding the bottleneck bandwidth and the inferred available bandwidth.

The bottleneck bandwidth characterizes the highest possible bandwidth between two sites. It is determined by the lowest link capacity provided on an end-to-end communication path and indicates at the technology deployed. Figure B.6 shows the histogram of the bottleneck bandwidth measured by Paxson. Obviously, most of the bottlenecks are determined by the underlying access technology of the sites. Paxson's analysis of the available bandwidth is derived from packet delay measurements.

Figure B.7(a) shows the cumulative density function (cdf) of inferred available bandwidth for a single bottleneck bandwidth value. It shows that for different bottleneck links having the same capacity the actual available bandwidth differs. This is reasonable since each link is quite probably not equally populated, that is, the number of competing flows on each link varies. A quite interesting results of the study is the fact that only a single cdf is provided. Paxson observed that the cdf holds for all bottleneck bandwidths with only very slight variations. As a result, the available bandwidth for all measured connections is then distributed as depicted in Figure B.7(b).

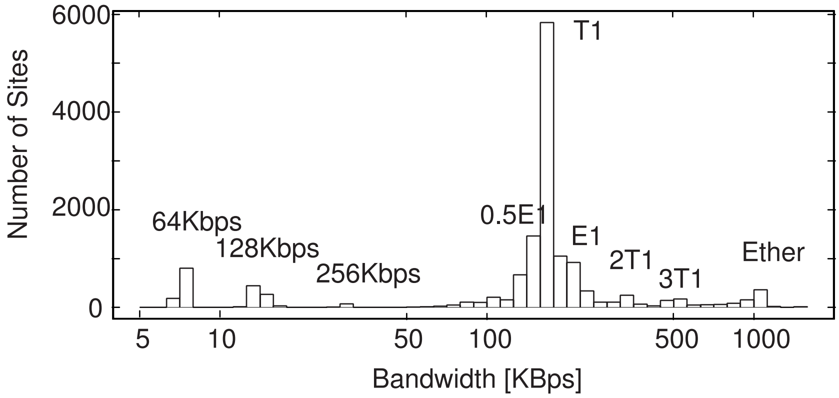


Figure B.6: Histogram of the measured bottleneck bandwidths.

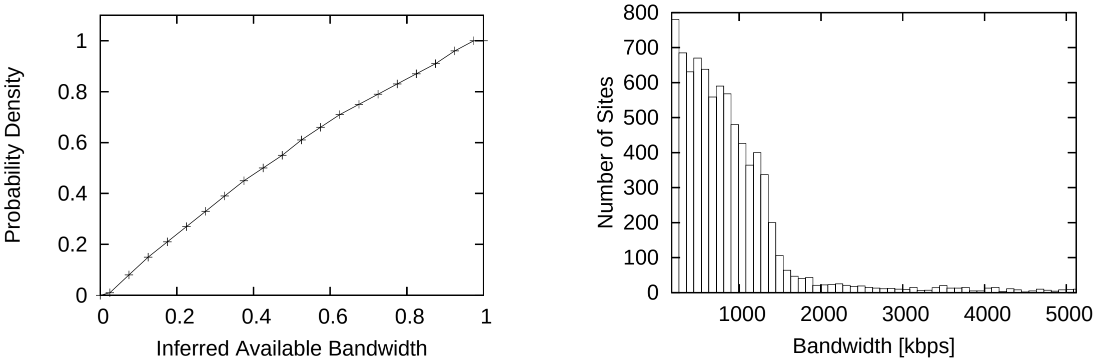


Figure B.7: Inferred cumulative density of available bandwidth and derived distribution of the available bandwidth.

B.5.2 PingER Measurement

The PingER (Ping End-to-end Reporting) project is conducted at the Stanford Linear Accelerator Center (SLAC). More than 300 sites worldwide are involved in the project, most of which are located at research institutions. The aggregate data of each month is publicly available.

Measurement in the PingER project is based on the Internet Control Message Protocol (ICMP) echo mechanism commonly referred to as “ping”. More specifically, sending pings and receiving the corresponding echoes allows for measurement of the round-trip time and the packet loss rate. Based on the measurement results, the simple TCP throughput formula (see Equation 5.2) is used to estimate the fair share distribution (see Figure B.5(b)).

Appendix C

Adaptive Rate Control Framework

For the purpose of a systematic approach to study the performance of the algorithms and mechanisms in Chapter 5 and Chapter 6, we developed the *Adaptive Rate Control Framework (ARCF)*. The modular and flexible design of ARCF provides a powerful tool for multicast as well as unicast control protocol research. The framework is available as a *ns-2* extension from `http://www.kom.tu-darmstadt.de/arcf`.

C.1 Design Overview

A class diagram of the ARCF implementation in *ns-2* [NS2] is presented in Figure C.1. We derived the base agent `ARCF Agent` from the *ns-2* base class `Agent` and derived a class hierarchy such that the `ARCF Source Agents` and `ARCF Sink Agents` are separated. A key design feature that makes the framework very flexible is the decoupling of the four components crucial to equation-based fair share estimation: (1) round-trip time estimator; (2) retransmission timeout estimator; (3) TCP formula; (4) loss estimator.

C.2 Equation-based Layered Multicast (ELM)

For the experiments in Chapter 5, modules have been implemented that provide the functionalities of TFRC as specified by the IETF [HFPW03]. For the experiments in Chapter 6, however, we implemented a multicast scheme that we refer to as

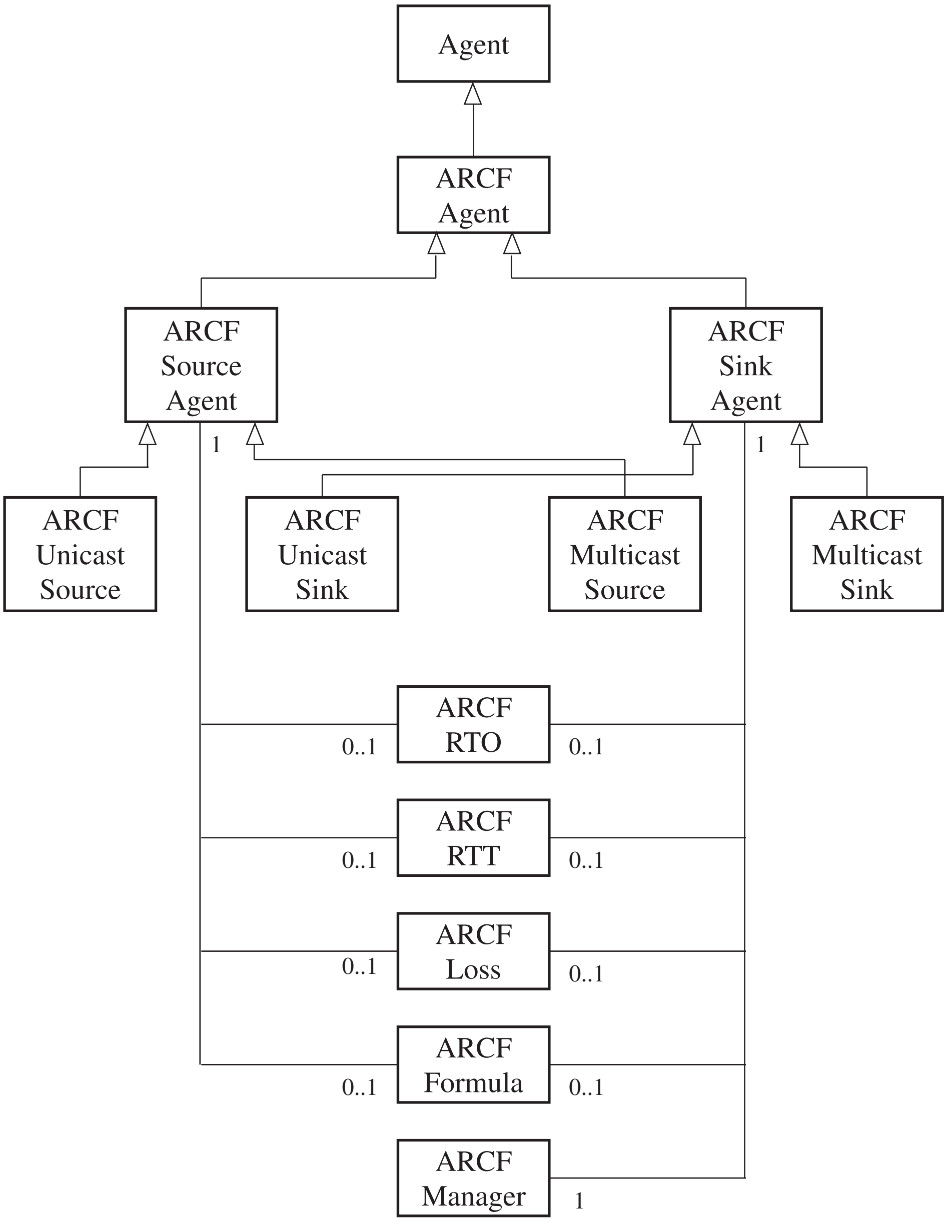


Figure C.1: Overview of the ARCF design: Class diagram.

Equation-Based Layered Multicast (ELM):

1. The round-trip time estimator uses a combination of one-way open-loop measurements and scalable closed-loop measurements. For that purpose, we implemented a mechanism proposed by Sisalem and Wolisz [SW00]. Finally,

the values are smoothed in order to provide an exponentially weighted moving average.

2. The loss estimator has been implemented based on our proposed extension for the original loss estimation algorithm. Hence, loss events as well as loss impacts are used for computation of the estimate as described in Chapter 6.

3. The group or subscription manager instantiates our lazy strategy with the sensitivity function discussed in Chapter 6. It is extended by an additional start-up mechanism in order to ensure convergence of ELM. The concept of the mechanism is briefly summarized subsequently.

Start-Up Mechanism

In the discussion in Section 5.6.3, we highlight the limitations of the TCP-model based approach to fair share estimation in low-multiplexing environments and the consequences for an equation-based layered multicast control scheme.

Particularly, the issue occurs when an corresponding receiver competes for resources on a bandwidth-limited bottleneck link with only a few TCP flows. At start-up phase, the receiver joins the base layer and is expected to gradually subscribe to a higher layers until converging to a fair subscription level. However, the above scenario might drive the receiver to calculate a very conservative estimate, thus, underestimating its fair level and potentially hindering it from joining the next layer.

To tackle this issue, an ELM receiver has available a second operational mode for the start-up phase. The start-up mechanism at the beginning of a receiver's participation in a multicast session attempts to force it to converge to a fair level. In that phase, the receiver operates as follows:

1. Join the base layer and wait for a time $\tau_{start-up}$.

2. Join the next layer and wait until a *deaf period* $\tau_{deaf} = \tau_{start-up} \cdot \frac{g_l}{g_1}$ elapses.

3. Compare the estimated fair share χ with the current group rate g_l:
    (a) if $\chi < g_l$, terminated start-up phase and switch to standard mode,
    (b) if $\chi \geq g_l$, proceed with step 2.

Author's Publications

Patents (Applications)

1. Ivica Rimac, José Rey, Rolf Hakenberg, Ralf Bauer, and Akito Fukui. Deterministic Feedback Collection for 3GPP MBMS. *European Patent Application 04020655.9*, August 2004.

2. Ivica Rimac, José Rey, Rolf Hakenberg, and Ralf Bauer. Feedback Control for Multicast or Broadcast Services. *European Patent Application 04018707.2*, August 2004.

3. Ivica Rimac, José Rey, and Rolf Hakenberg. Adaptive and Scalable QoS Architecture for Multi-bearer Multicast/Broadcast Services. *European Patent Application 04014513.8*, June 2004.

4. Ivica Rimac, José Rey, and Rolf Hakenberg. Adaptive and Scalable QoS Architecture for Single-bearer Multicast/Broadcast Services. *European Patent Application 04014494.1*, June 2004.

5. Ivica Rimac, José Rey, and Rolf Hakenberg. Delivery Mechanism for Static Media Objects. *U.S. Patent Application 10/817,965, German Utility Model 20-2004-005445.0*, April 2004.

Journal Articles

6. Andreas Faatz, Adulmotaleb El Saddik, Stefan Hoermann, Ivica Rimac, Cornelia Seeberg, Achim Steinacker, and Ralf Steinmetz. Multimedia und Wissen: Unser Weg zu einem produktiven Umgang mit Wissensdurst. *thema Forschung*, 2000(2):26-33, November 2000.

Conference Contributions

7. Matthias Hollick, Ivan Martinovic, Tronje Krop, and Ivica Rimac. A Survey on Dependable Routing in Sensor Networks, Ad hoc Networks, and Cellular Networks. In *Proceedings of the 30th Euromicro Conference (to be published)*, Rennes, France, August 2004.

8. Ivica Rimac, Jens Schmitt, and Ralf Steinmetz. Observations on Equation-based Estimation of TCP-compatible Rate for Multi-rate Multicast Scenarios. In *Proceedings of the International Workshop on Multimedia Interactive Protocols and Systems (MIPS 2003)*, pages 66–77, Naples, Italy, November 2003.

9. Ivica Rimac, Wolfram Liese, Jens Schmitt, and Ralf Steinmetz. Equation-Based Approach to TCP-Compatible Multicast Congestion Control for Layered Transmission in Low-Multiplexing Environments. In *Proceedings of the 22nd IEEE International Performance Computing and Communications Conference (IPCCC 2003)*, pages 469–473, Phoenix, AZ, USA, April 2003.

10. Ivica Rimac, Jens Schmitt, and Ralf Steinmetz. Is Dynamic Multi-Rate Multicast Worthwhile the Effort? In *Proceedings of the 28th Euromicro Conference 2002 (Multimedia and Telecommunications Track)*, pages 233–239, Dortmund, Germany, September 2002.

11. Achim Steinacker, Andreas Faatz, Cornelia Seeberg, Ivica Rimac, Stefan Hoermann, Abdulmotaleb El Saddik, and Ralf Steinmetz. MediBook: Combining Semantic Networks with Metadata for Learning Resources to Build a Web Based Learning System. In *Proceedings of the Conference on Educational Multimedia, Hypermedia & Telecommunications (EdMedia 2001)*, Tampere, Finnland, June 2001.

12. Cornelia Seeberg, Ivica Rimac, Stefan Hoermann, Andreas Faatz, Achim Steinacker and Abdulmotaleb El Saddik, and Ralf Steinmetz. MediBook: Realisierung eines generischen Ansatzes für ein internetbasiertes Multimedia-Lernsystem am Beispiel Medizin. In *Tagungsband: Treffen der GI-Fachgruppe 1.1.3 Maschinelles Lernen (GMD Report 114)*, pages 96–105, September 2000.

13. Ivica Rimac, Stephan Fischer, and Ralf Steinmetz. The ICOR Framework: A Top-down Approach to Media Indexing and Retrieval. In *Proceedings of*

the IEEE International Conference on Multimedia and Expo (ICME 2000), pages 149–152, New York, NY, USA, July 2000.

14. Jana Dittmann, Martin Steinebach, Ivica Rimac, Stephan Fischer, and Ralf Steinmetz. Combined video and audio watermarking: Embedding content information in multimedia data. In *Proceedings of SPIE: Security and Watermarking of Multimedia Contents II*, volume 3971, pages 455–464, San Jose, CA, USA, January 2000.

15. Ivica Rimac, Lutz Finsterle, Stephan Fischer, and Ralf Steinmetz. Music retrieval in ICOR. In *Proceedings of SPIE: Multimedia Storage and Archiving Systems IV*, volume 3846, pages 90–101, Boston, MA, USA, September 1999.

16. Stephan Fischer, Ivica Rimac, and Ralf Steinmetz. Interactive Panning and Zooming the Virtual Camera. In *Proceedings of IEEE International Conference on Multimedia Computing and Systems (ICMCS 1999)*, pages 138-142, Florence, Italy, June 1999.

17. Ivica Rimac, Stephan Fischer, and Ralf Steinmetz. Automatic Recognition of Camera Zooms. In *Proceedings of Visual 99*, pages 253-260, Amsterdam, Netherlands, June 1999.

Curriculum Vitae

Personal Details

Name:	Ivica Rimac
Date of birth:	09.11.1971
Marital status:	Married, one child

Education

1991 – 1997	Darmstadt University of Technology Department of Electrical Engineering & Information Technology Degree: Diplom-Ingenieur (Dipl.-Ing.)
1988 – 1991	Berufliches Gymnasium (High School) Gewerblich-Technische-Schulen, Offenbach Degree: Abitur
1978 – 1988	Elementary School and Junior High School, Oberthausen

Professional Experience

01/1999 – present	Research Assistant Multimedia Communications Lab (KOM) Darmstadt University of Technology
01/1997 – 12/1998	Graduate Assistant (part-time) Institute for System and Process Communication (KOM) Darmstadt University of Technology
11/1997 – 02/2001	Freelancer as IT-Professional and IT-Trainer Web design, web programming, and Java programming

Supervised Student, Diploma, and Master Theses

1. Gian Donato Colussi. Equation-Based Layered Multicast Congestion Control. *Master Thesis KOM–D–215*, September 2004.

2. Markus Hahn. A Scalable Feedback Mechanism for the Collection of Receiver Bandwidth Capabilities in a Multicast Session. *Student Thesis KOM–S–175*, September 2004.

3. André Beyer. Adaptive Multimedia-Multicast-Dienste in 3G Netzen. *Student Thesis KOM–S–174*, August 2004.

4. Alexander Bourgett. Context Transfer Based Mechanisms for Mobility Improvement in 3G-WLAN Interworking. *Diploma Thesis KOM–D–202*, in collaboration with the Panasonic European Laboratories GmbH, Langen, November 2003.

5. Alexandre Bouffier. Color Normalization for the Robust Recognition of Video Sequences. *Student Thesis KOM–S–169*, September 2003.

6. Matthias Wengenroth. Multicast Streaming Szenario in 3G. *Diplomarbeit KOM–D–190*, March 2003

7. Wolfram Liese. Implementierung und Evaluierung eines Protokolls fr überlastkontrolliertes Layered Multicast. *Diploma Thesis KOM–D–185*, January 2003.

8. José Miguel Cardona Pastor. Simulation-based Evaluation of Multicast Congestion Control Mechanisms. *Master Thesis KOM–D–175*, November 2002.

9. Wolfram Liese. Untersuchung von sender- und empfängerseitigen Mechanismen für überlastkontrolliertes dynamisches Layered Multicast. *Student Thesis KOM–S–130*, September 2002.

10. Gita Gharazoglou. Layered Video: Kodierung, Übertragung und Unterstützung in MPEG-4. *Student Thesis KOM–S–116*, July 2001.

11. Hugo Ansgar. Analyse von Eigenschaften und Inhaltsmerkmalen aus Audiodaten und Implementierung einer spezialisierten Java-Applikation. *Student Thesis KOM–S–096*, in collaboration with the Gesellschaft für Mathematik und Datenverarbeitung mbH (GMD), Darmstadt,October 2000.

12. Shirin Ahmadzadeh Shad. Entwurf und Implementierung eines XML-basierten Topic-Maps-Navigators. *Diploma Thesis KOM–D–120*, August 2000.
13. Stefan Hörmann. Automatische Generierung von Präsentationen im MediBook. *Diploma Thesis KOM–D–115*, July 2000.
14. Peter Beck. Implementierung eines flexiblen Algorithmus zum Tracking von Objekten in Java. *Student Thesis KOM–S–053*, January 2000.
15. Miriam Leon Cristobal. Vergleich von Verfahren zur Bestimmung von Bewegungsvektoren in digitalem Video. *Diploma Thesis KOM–D–086*, December 1999.
16. Cristina Saraiva. Entwurf eines Konzepts zur semiautomatischen Indizierung audiovisueller Daten. *Student Thesis KOM–S–052*, October 1999.
17. Davor-Jurica Dobra. Indizierung von Sprachdokumenten: Ansatz und Evaluierung von Spracherkennungs-Software. *Diploma Thesis KOM–D–087*, September 1999.
18. Stefan Hörmann. Automatische Gesichtsentdeckung in digitalen Bildsequenzen. *Student Thesis KOM–S–050*, September 1999.
19. Kerim Mansur. Entwicklung eines komponentenbasierten Java-Applikations-Frameworks für die Medienverarbeitung. *Diploma Thesis KOM–D–085*, August 1999.
20. Michael Schneider. Implementierung von Bildsegmentierungsalgorithmen in Java. *Diploma Thesis KOM–D–084*, June 1999.

Bibliography

[ABD01] Kevin C. Almeroth, Supratik Bhattacharyya, and Christophe Diot. Challenges of integrating ASM and SSM IP multicast protocol architectures. In *Proceedings of the International Workshop on Digital Communications: Evolutionary Trends of the Internet (IWDC'01)*, Taormina, Italy, September 2001.

[AMV96] Elan Amir, Steve McCanne, and Martin Vetterli. A layered DCT coder for Internet video. In *Proceedings of the IEEE International Conference on Image Processing (ICIP) 1996*, pages 13–16, Lausanne, Switzerland, September 1996.

[Bal97] Antonio Ballardie. Core based trees (CBT) multicast routing architecture. Internet Engineering Task Force (IETF), RFC 2201, September 1997.

[BBFS01] Deepak Bansal, Hari Balakrishnan, Sally Floxd, and Scott Shenker. Dynamic behavior of slowly-responsive congestion control algorithms. In *Proceedings of ACM SIGCOMM 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 263–274, San Diego, CA, USA, August 2001.

[BBK02] Suman Banerjee, Bobby Bhattacharjee, and Christopher Kommareddy. Scalable application layer multicast. In *Proceedings of ACM SIGCOMM 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 205–217, Pittsburgh, PA, USA, August 2002.

[BCC+98] Bob Braden, David D. Clark, Jon Crowcroft, Bruce Davie, Steve Deering, Deborah Estrin, Sally Floyd, Van Jacobson, Greg Minshall, Craig Partridge, Larry Peterson, K. K. Ramakrishnan, Scott Shenker, John Wroclawski, and Lixia Zhang. Recommendations on queue management and congestion avoidance in the Internet. Internet Engineering Task Force (IETF), RFC 2309, April 1998.

[BCHC02] Ian Brown, John Crowcroft, Mark Handley, and Brad Cain. Internet multicast tomorrow. *The Internet Protocol Journal*, 5(4):2–19, December 2002.

[BFH$^+$00] John Byers, Michael Frumin, Gavin Horn, Michael Luby, Michael Mitzenmacher, Alex Roetter, and William Shaver. FLID-DL: Congestion control for layered multicast. In *Proceedings of the Second International Workshop on Networked Group Communication (NGC 2000)*, pages 71–81, Stanford, CA, USA, November 2000.

[BHH$^+$00] John Byers, Mark Handley, Gavin Horn, Michael Luby, and Lorenzo Vicisano. More thoughts on reference simulations for reliable multicast congestion control schemes. Meeting notes, Digital Fountain, Inc., August 2000.

[BHL$^+$02] John W. Byers, Gavin Horn, Michael Luby, Michael Mitzenmacher, and William Shaver. FLID-DL: Congestion control for layered multicast. *IEEE Journal on Selected Areas in Communications (JSAC): Network Support for Multicast Communication*, 20(8):1558–1570, October 2002.

[BM01] Thomas Bonald and Laurent Massoulié. Impact of fairness on internet performance. In *Proceedings of ACM SIGMETRICS 2001 Joint International Conference on Measurement and Modeling of Computer Systems*, pages 82–91, Cambridge, MA, USA, June 2001.

[BTW94] Jean-Chrysostome Bolot, Thierry Turletti, and Ian Wakeman. Scalable feedback control for multicast video distribution in the internet. In *Proceedings of ACM SIGCOMM '94 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 58–67, London, UK, October 1994.

[CDF$^+$02] Brad Cain, Steve Deering, Bill Fenner, Isidor Kouvelas, and Ajit Thyagarajan. Internet group management protocol, version 3. Internet Engineering Task Force (IETF), RFC 3376, October 2002.

[CH01] Zhibo Chen and Yun He. Quality weighted bit allocation for smoother streaming of stored FGS video. In *Proceedings of International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2001)*, Las Vegas, NV, USA, June 2001.

[Cha03] Yatin Chawathe. Scattercast: An adaptable broadcast distribution framework. *ACM/Springer Multimedia Systems Journal*, 9(1):104–118, July 2003.

[CLN04] Yi Cui, Baochun Li, and Klara Nahrstedt. oStream: Asynchronous streaming multicast in application-layer overlay networks. *IEEE Journal on Selected Areas in Communications (JSAC), Special Issue on Recent Advances in Service Overlays*, 22(1):91–106, January 2004.

[CMB00] Yatin Chawathe, Steven McCanne, and Eric Brewer. RMX: Reliable multicast in heterogeneous environments. In *Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000)*, volume 2, pages 795–804, Tel-Aviv, Israel, March 2000.

[CN03] Yi Cui and Klara Nahrstedt. Proxy-based asynchronous multicast for efficient on-demand media distribution. In *Proceedings of the SPIE/ACM Conference on Multimedia Computing and Networking (MMCN 2003)*, Santa Clara, CA, USA, January 2003.

[Col04] Gian Donato Colussi. Equation-based layered multicast. Master's thesis, Darmstadt University of Technology, Darmstadt, Germany, September 2004.

[CRSZ01] Yang-Hua Chu, Sanjay G. Rao, Srinivasan Seshan, and Hui Zhang. Enabling conferencing applications on the internet using an overlay multicast. In *Proceedings of ACM SIGCOMM 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 55–67, San Diego, CA, USA, August 2001.

[CS02] Philip A. Chou and Anshul Seghal. Rate-distortion optimized receiver-driven streaming over best-effort networks. In *Proceedings of the 12th International Packet Video Workshop (PV 2002)*, Pittsburg, PA, USA, April 2002.

[CT91] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991. ISBN 0471062596.

[CT00] Brad Cain and Donald F. Towsley. Generic multicast transport services: Router support for multicast applications. In *Proceedings of the IFIP-TC6/European Commission Internation Conference NETWORKING 2000: Broadband Communications, High Performance Networking, and Performance of Communication Networks*, volume 1815 of *Lecture Notes in Computer Science*, pages 108–120, Paris, France, May 2000. Springer-Verlag.

[CZ97] Ti-Hao Chiang and Ya-Qin Zhang. A new rate control scheme using quadratic distortion mode. *IEEE Transactions on Circuits and Systems for Video Technology (CSVT)*, 7(1):246–250, November 1997.

[Dan89] Peter B. Danzig. *Optimally Selecting the Parameters of Adaptive Backoff Algorithms for Computer Networks and Multiprocessors.* PhD thesis, University of California at Berkeley, Berkeley, CA, USA, December 1989.

[dCRR02] Philippe de Cuetos, Martin Reisslein, and Keith W. Ross. Analysis of a large library of MPEG-4 FGS rate-distortion traces for streaming video. Technical Report RR-02-068, Institut Eurocom, Sophia Antipolis, France, June 2002.

[dCRR03] Philippe de Cuetos, Martin Reisslein, and Keith W. Ross. Evaluating the streaming of FGS-encoded video with rate-distortion traces. Technical Report RR-03-078, Institut Eurocom, Sophia Antipolis, France, June 2003.

[DDC97] Christophe Diot, Walid Dabbous, and Jon Crowcroft. Multipoint communication: A survey of protocols, functions and mechanisms. *IEEE Journal on Selected Areas in Communications (JSAC)*, pages 277–290, April 1997.

[Dee89] Steve E. Deering. Host extensions for ip multicasting. Internet Engineering Task Force (IETF), RFC1112, August 1989.

[Den00] Robert Denda. The fairness challenge in computer networks. Technical Report 6/2000, University of Mannheim, Praktische Informatik IV, D-68131 Mannheim, 2000.

[DGS+03] Christophe Diot, Leonard Giuliano, Greg Shepherd, Robert Rockell, David Meyer, John Meylor, and Brian Haberman. An overview of source-specific multicast (SSM). Internet Engineering Task Force (IETF), RFC 3569, July 2003.

[DL03] Min Dai and Dmitri Loguinov. Analysis of rate-distortion functions and congestion control in scalable internet video streaming. In *Proceedings of the 13th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 2003)*, pages 60–69, Monterey, CA, USA, June 2003.

[DLL+00] Christophe Diot, Brian Neil Levine, Bryan Lyles, Hassan Kassem, and Doug Balensiefen. Deployement issues for the IP multicast service and architecture. *IEEE Network: Special Issue on Multicasting*, pages 78–88, January/February 2000.

[DLR03] Min Dai, Dmitri Loguinov, and Hayder Radha. Analysis and distortion modeling of MPEG-4 FGS. In *Proceedings of the IEEE International Conference on Image Processing (ICIP 2003)*, volume 3, pages 301–304, Barcelona, Spain, September 2003.

[DO97] Dante DeLucia and Katia Obraczka. Multicast feedback suppression using representatives. In *Proceedings of the Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '97)*, volume 2, pages 463–470, Kobe, Japan, April 1997.

[EFH+98] Deborah Estrin, Dino Farinacci, Ahmed Helmy, David Thaler, Stephen Deering, Mark Handley, Van Jacobson, Ching-Gung Liu, Puneet Sharma, and Liming Wei. Protocol independent multicast-sparse mode (PIM-SM): Protocol specification. Internet Engineering Task Force (IETF), RFC 2326, April 1998.

[FH99] Sally Floyd and Tom Henderson. The NewReno modification to TCP's fast recovery algorithm. Internet Engineering Task Force (IETF), RFC 2518, April 1999.

[FHPW00] Sally Floyd, Mark Handley, Jitendra Padye, and Jörg Widmer. Equation-based congestion control for unicast applications. In *Proceedings of ACM SIGCOMM 2000 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 43–56, Stockholm, Sweden, August 2000.

[FJ92] Sally Floyd and Van Jacobson. On traffic phase effects in packet-switched gateways. *Internetworking: Research and Experience*, 3(3):115–156, September 1992.

[FJ93] Sally Floyd and Van Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking (TON)*, 1(4):397–413, August 1993.

[FJL+97] Sally Floyd, Van Jacobson, Ching-Gung Liu, Steven McCanne, and Lixia Zhang. A reliable multicast framework for light-weight sessions and application level framing. *IEEE/ACM Transactions on Networking (TON)*, 5(6):784–803, December 1997.

[FKP04] Sally Floyd, Eddie Kohler, and Jitendra Padhye. Profile for DCCP congestion control ID 3: TFRC congestion control. Internet Engineering Task Force (IETF), Internet Draft draft-ietf-dccp-ccid3-06, July 2004.

[Flo91] Sally Floyd. Connections with multiple congested gateways in packet-switched networks part 1: One-way traffic. *ACM SIGCOMM Computer Communication Review*, 21(5):30–47, October 1991.

[Flo94] Sally Floyd. TCP and explicit congestion notification. *ACM Computer Communication Review*, 24(5):10–23, October 1994.

[Flo00] Sally Floyd. Congestion control principles. Internet Engineering Task Force (IETF), RFC 2914, September 2000.

[FML^{+}03] Chuck Fraleigh, Sue Moon, Bryan Lyles, Chase Cotton, Mujahid Khan, Deb Moll, Rob Rockell, Ted Seely, and Christophe Diot. Packet-level traffic measurements from the sprint ip backbone. *IEEE Network*, 17(6):6–16, November–December 2003.

[FT99] Timur Friedman and Donald F. Towsley. Multicast session membership size estimation. In *Proceedings of the Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '99)*, volume 2, pages 965–972, New York, NY, USA, March 1999.

[GGHS99] Raman Gopalakrishnan, James Griffioen, Gisli Hjálmtýsson, and Cormac Sreenan. Stability and fairness issues in layered multicast. In *Proceedings of the 9th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '99)*, Basking Ridge, NJ, USA, June 1999.

[Gir88] Bernd Girod. Ein Modell der menschlichen visuellen Wahrnehmung zur Irrelevanzreduktion von Fernsehluminanzsignalen. *VDI-Fortschritt-Berichte*, 10(84), 1988.

[Gri00] Carsten Griwodz. *Wide-area True Video-on-Demand by a Decentralized Cache-based Distribution Infrastructure.* PhD thesis, Darmstadt University of Technology, Darmstadt, Germany, June 2000.

[Gro97] Matthias Grossglauser. Optimal deterministic timeouts for reliable scalable multicast. *IEEE Journal on Selected Areas in Communications (JSAC)*, 15(3):422–433, April 1997.

[GV01] Sergey Gorinsky and Harrick Vin. The utility of feedback in layered multicast congestion control. In *Proceedings of the 11th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 2001)*, pages 93–102, Port Jefferson, NY, USA, June 2001.

[HC99] Mark Handley and John Crowcroft. Internet multicast today. *The Internet Protocol Journal*, 2(4):2–19, December 1999.

[HFPW03] Mark Handley, Sally Floyd, Jitendra Padhye, and Jörg Widmer. TCP friendly rate control (TFRC): Protocol specification. Internet Engineering Task Force (IETF), RFC 3448, January 2003.

[Hof96] Markus Hofmann. A generic concept for large-scale multicast. In B. Plattner, editor, *Proceedings of the International Zurich Seminar on Digital Communication*, Lecture Notes in Computer Science, pages 95–106, Zurich, Switzerland, 1996. Springer-Verlag.

[ITU02] International Telecommunication Union – Radiocommunication. Methodology for the subjective assessment of the quality of television pictures. International Standard, ITU-R BT.500-11, June 2002.

[Jac94] Van Jacobson. Multimedia conferencing on the internet. Totorial at the ACM SIGCOMM '94 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, August 1994.

[JAZ98] Tianji Jiang, Mostafa H. Ammar, and Ellen W. Zegura. Inter-receiver fairness: A novel performance measure for multicast ABR sessions. In *Proceedings of the ACM SIGMETRICS 1998 Joint International Conference on Measurement and Modeling of Computer Systems*, pages 202–211, Madison, WI, USA, June 1998.

[JCH84] Rajendra K. Jain, Dah-Ming. W. Chiu, and William R. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer system. Technical Report DEC-TR-301, Digital Equipment Corporation, Eastern Research Lab, Hudson, MA, USA, September 1984.

[JGJ$^+$00] John Jannotti, David K. Gifford, Kirk L. Johnson, M. Frans Kaashoek, and James W. O'Toole, Jr. Overcast: Reliable multicasting with an overlay network. In *Proceedings of the Fourth Symposium on Operating System Design and Implementation (OSDI 2000)*, pages 197–212, San Diego, CA, USA, October 2000.

[JZA99] Tianji Jiang, Ellen W. Zegura, and Mostafa H. Ammar. Inter-receiver fair multicast communication over the internet. In *Proceedings of the 9th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '99)*, Basking Ridge, NJ, USA, June 1999.

[KB03] Gu-In Kwon and John W. Byers. Smooth multirate multicast congestion control. In *Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003)*, volume 2, pages 1022–1032, San Francisco, CA, USA, April 2003.

[KHF04] Eddie Kohler, Mark Handley, and Sally Floyd. Datagram congestion control protocol (DCCP). Internet Engineering Task Force (IETF), Internet Draft draft-ietf-dccp-spec-07, July 2004.

[KHTK00] Sneha Kumar Kasera, Gísli Hjálmtýsson, Donald F. Towsley, and James F. Kurose. Scalable reliable multicast using multiple multicast channels. *IEE/ACM Transactions on Networking (TON)*, 8(3):294–310, June 2000.

[Kos98] Dave Kosiur. *IP Multicast*. Wiley Computer Publishing, 1998. ISBN 0-471-24359-0.

[KTPE99] Jun-ichi Kimura, Fouad A. Tobagi, Jose-Miguel Pulido, and Peder J. Emstad. Perceived quality and bandwidth characterization of layered MPEG-2 video encoding. In *Proceedings of the SPIE International Symposium on Voice, Video and Data Communications*, Boston, MA, USA, September 1999.

[LB82] F. J. Lukas and Zigmantas L. Budrikis. Picture quality prediction based on a visual model. *IEEE Transactions on Communications*, 30(7):1679–1692, July 1982.

[LB99] Jörg Liebeherr and Tyler K. Beam. HyperCast: A protocol for maintaining multicast group members in a logical hypercube topology. In *Proceedings of the First International Workshop on Networked Group Communication (NGC '99)*, pages 72–89, Pisa, Italy, November 1999.

[LB00a] Arnaud Legout and Ernst W. Biersack. Pathological behaviors for RLM and RLC. In *Proceedings of the 10th International Workshop on Networking and Operating Systems Support for Digital Audio and Video (NOSSDAV 2000)*, Chapel Hill, NC, USA, June 2000.

[LB00b] Arnaud Legout and Ernst W. Biersack. PLM: Fast convergence for cumulative layered multicast transmission schemes. In *Proceedings of the ACM SIGMETRICS 2000 Joint International Conference on Measurement and Modeling of Computer Systems*, pages 13–22, Sant Clara, CA, USA, June 2000.

[LGSH02] Michael Luby, Vivek K Goyal, Simon Skaria, and Gavin B. Horn. Wave and equation based rate control using multicast round trip time. In *Proceedings of ACM SIGCOMM 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 191–204, Pittsburgh, PA, USA, August 2002.

[LGT98] Li-Wei H. Lehman, Stephen J. Garland, and David L. Tennenhouse. Active reliable multicast. In *Proceedings*, volume 2, pages 581–589, San Francisco, CA, USA, March 1998.

[LHL00] Cheng Liang, Ng Chee Hock, and Zhang Liren. Feedback control using representatives and timer in reliable multicast protocols. In *IEEE TENCON 2000*, volume 1, pages 338–340, Kuala Lumpur, Malaysia, September 2000.

[Li01] Weiping Li. Overview of fine granularity scalability in MPEG-4 video standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(3):301–317, March 2001.

[LLZ02] Jiangchuan Liu, Bo Li, and Ya-Qin Zhang. A hybrid adaptation protocol for TCP-friendly layered multicast and its optimal rate allocation. In *Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, volume 3, pages 1520–1529, New York, NY, USA, June 2002.

[LM97] T. V. Lakshman and Upamanyu Madhow. The performance of TCP/IP for networks with high bandwidth-delay products and random loss. *IEEE/ACM Transactions on Networking (TON)*, 5(3):336–350, June 1997.

[LNB01] Arnaud Legout, Jörg Nonnenmacher, and Ernst W. Biersack. Bandwidth-allocation policies for unicast and multicast flows. *IEEE/ACM Transactions on Networking (TON)*, 9(4):464–478, August 2001.

[LPA98] Xue Li, Sanjoy Paul, and Mustafa Ammar. Layered video multicast with retransmissions (LVMR): Evaluation of hierarchical rate control. In *Proceedings of the Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '98)*, volume 3, pages 1062–1072, San Francisco, CA, USA, March 1998.

[Mau98] Andreas Ulrich Mauthe. *End-to-End Support for Multimedia Multipeer Communications.* PhD thesis, Lancaster University, Lancaster, UK, November 1998.

[MC00] Warren Matthews and Les Cottrell. The pinger project: Active internet performance monitoring for the henp community. *IEEE Communications Magazine on Network Traffic Measurement and Experiments*, 38(5):130–136, May 2000.

[MJV96] Steven McCanne, Van Jacobson, and Martin Vetterli. Receiver-driven layered multicast. In *Proceedings of ACM SIGCOMM '96 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 117–130, Stanford, CA, USA, August 1996.

[MMFR96] Matt Mathis, Jamshid Mahdavi, Sally Floyd, and Allyn Romanow. TCP selective acknowledgement options. Internet Engineering Task Force (IETF), RFC 2018, October 1996.

[Moh03] Samir Mohamed. *Automatic Evaluation of Real-Time Multimedia Quality: A Neural Network Approach.* PhD thesis, University of Rennes I, Rennes, France, 2003.

[Moy94] John Moy. Multicast extensions to OSPF. Internet Engineering Task Force (IETF), RFC 1584, March 1994.

[MSMO97] Matthew Mathis, Jeffrey Semke, Jamshid Mahdavi, and Teunis Ott. The macroscopic behavior of the TCP congestion avoidance algorithm. *ACM SIGCOMM Computer Communication Review*, 27(3):67–82, July 1997.

[NB98] Jörg Nonnenmacher and Ernst W. Biersack. Optimal multicast feedback. In *Proceedings of the Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '98)*, volume 3, pages 964–971, San Francisco, CA, USA, March 1998.

[NS2] The network simulator ns-2. http://www.isi.edu/nsnam/ns/.

[Obr98] Katia Obraczka. Multicast transport protocols: A survey and taxonomy. *IEEE Communications Magazine*, 36(1):94–102, January 1998.

[Pad00] Jitendra D. Padhye. *Towards a Comprehensive Congestion Control Framework for Continuous Media Flows in Best Effort Networks.* PhD thesis, University of Massachusetts Amherst, Amherst, MA, USA, March 2000.

[Pap01] Lothar Papula. *Mathematik für Ingenieure und Naturwissenschaftler*, volume 2. Vieweg-Verlag, 10th edition, 2001. ISBN 3528942371.

[Pax97a] Vern Paxson. Automated packet trace analysis of TCP implementations. In *Proceedings of the ACM SIGCOMM '97 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, Cannes, France, August 1997.

[Pax97b] Vern Paxson. End-to-end internet packet dynamics. *ACM Computer Communication Review*, 27(4), October 1997.

[Pax97c] Vern Paxson. End-to-end internet packet dynamics. In *Proceedings of the ACM SIGCOMM '97 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, Cannes, France, August 1997.

[PE02] Fernando Pereira and Touradj Ebrahimi. *The MPEG-4 Book*. Prentice Hall PTR, Upper Saddle River, NJ, 2002. ISBN 0-13-061621-4.

[PF01] Jitendra Padhye and Sally Floyd. On inferring TCP behavior. In *Proceedings of ACM SIGCOMM 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 287–298, San Diego, CA, USA, August 2001.

[PFTK98] Jitendra Padhye, Victor Firoiu, Don Towsley, and Jim Kurose. Modeling TCP throughput: A simple model and its empirical validation. In *Proceedings of ACM SIGCOMM '98 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 303–314, Vancouver, BC, Canada, September 1998.

[Pos81] Jon Postel. Internet protocol. Internet Engineering Task Force (IETF), RFC 791, September 1981.

[PR02] Pragyansmita Paul and S. V. Raghavan. Survey of multicast routing algorithms and protocols. In *Proceedings of the 15th International Conference on Computer Communication*, pages 902–926, Mulmbai, Maharashtra, India, August 2002.

[PSLB97] Sanjoy Paul, Krishan K. Sabnani, John C.-H. Lin, and Supratik Bhattacharyya. Reliable multicast transport protocol (RMTP). *IEEE Journal on Selected Areas in Communications (JSAC)*, 15(3):407–421, April 1997.

[PW03] Margaret H. Pinson and Stephen Wolf. Comparing subjective video quality testing methodologies. In Touradj Ebrahimi and Thomas Sikora, editors, *Proceedings of SPIE Visual Communications and Image Processing 2003*, volume 5150, pages 573–582, June 2003.

[RFB01] K. K. Ramakrishnan, Sally Floyd, and David L. Black. The addition of explicit congestion notification (ECN) to IP. Internet Engineering Task Force (IETF), RFC 3168, September 2001.

[Riz00] Luigi Rizzo. PGMCC: A TCP-friendly single-rate multicast congestion control scheme. In *Proceedings of ACM SIGCOMM 2000 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 17–28, Stockholm, Sweden, August 2000.

[RKT02] Dan Rubenstein, Jim Kurose, and Don Towsley. The impact of multicast layering on network fairness. *IEEE/ACM Transactions on Networking (TON)*, 10(2):169–182, April 2002.

[RLSS03] Ivica Rimac, Wolfram Liese, Jens Schmitt, and Ralf Steinmetz. Equation-based approach to TCP-compatible multicast congestion control for layered transmission in low-multiplexing environments. In *Proceedings of the 22nd IEEE International Performance Computing and Communications Conference (IPCCC 2003)*, pages 469–473, Phoenix, AZ, USA, April 2003.

[RR99] Sridhar Ramesh and Injong Rhee. Issues in TCP model-based flow control. Technical report tr-99-15, North Carolina State University, Raleigh, NC, USA, 1999.

[RSS03] Ivica Rimac, Jens Schmitt, and Ralf Steinmetz. Observations on equation-based estimation of TCP-compatible rate for multi-rate multicast scenarios. In *Proceedings of the International Workshop on Multimedia Interactive Protocols and Systems (MIPS 2003)*, volume 2899, pages 66–77, Naples, Italy, November 2003. ISBN 3-540-20534-9.

[RYHE00] Reza Rejaie, Haobo Yu, Mark Handley, and Deborah Estrin. Multimedia proxy caching mechanism for quality adaptive streaming applications in the Internet. In *Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000)*, volume 2, pages 980–989, Tel Aviv, Israel, March 2000.

[SA00] Jamal Hadi Salim and Uvaiz Ahmed. Performance evaluation of explicit congestion notification (ECN) in IP networks. Internet Engineering Task Force (IETF), RFC 2884, July 2000.

[Sac02] Lothar Sachs. *Angewandte Statistik*. Springer-Verlag, 2002. ISBN 3540424482.

[SCFJ03] Henning Schulzrinne, Stephen L. Casner, Ron Frederick, and Van Jacobson. RTP: A transport protocol for real-time applications. Internet Engineering Task Force (IETF), RFC 3550, July 2003.

[Sch01] Jens Burkhard Schmitt. *Heterogeneous Network Quality of Service Systems.* Kluwer Academic Publishers, 2001. ISBN 0-793-7410-X.

[Sha92] Nachum Shacham. Multipoint communication by hierarchically encoded data. In *Proceedings of the Eleventh Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 1992)*, volume 3, pages 2107–2114, Florence, Italy, May 1992.

[She95] Scott Shenker. Fundamental design issues for the future internet. *IEEE Journal on Selected Areas in Communications (JSAC)*, 13(7):1176–1188, September 1995.

[SN04] Ralf Steinmetz and Klara Nahrstedt. *Multimedia Systems.* Springer-Verlag, 2004. ISSN 1612-1449.

[ST00] Saswati Sarkar and Leandros Tassiulas. Fair allocation of discrete bandwidth layers in multicast networks. In *Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000)*, volume 3, pages 1491–1500, Tel-Aviv, Israel, March 2000.

[Str02] Tilo Strutz. *Bilddatenkompression.* Vieweg-Verlag, 2nd edition, Februar 2002. ISBN 3-528-13922-6.

[SW00] Dorgham Sisalem and Adam Wolisz. MLDA: A TCP-friendly congestion control framework for heterogeneous multicast environments. In *Proceedings of the 8th International Workshop on Quality of Service (IWQoS 2000)*, Pittsburgh, USA, June 2000.

[TGP98] K. T. Tan, Mohammed Ghanbari, and Don E. Pearson. An objective measurement tool for MPEG video quality. *Signal Processing: Special Issue on Image and Video Quality Metrics*, 70(3):279–294, November 1998.

[Tur94] Thierry Turletti. The INRIA videoconferencing system (IVS). *ConneXions – The Interoperability Report Journal*, 8(10):20–24, October 1994.

[UKB02] Guillaume Urvoy-Keller and Ernst W. Biersack. A multicast congestion control model for overlay networks and its performance. In *Proceedings* *of the Fourth International Workshop on Networked Group Communications (NGC 2002)*, Boston, MA, USA, October 2002.

[VQE00] Video Quality Expert Group. Final report from the Video Quality Experts Group on the validation of objective models of video quality measurement, March 2000. available at *http://www.vqeg.org.*

[VRC98] Lorenzo Vicisano, Luigi Rizzo, and Jon Crowcroft. TCP-like congestion control for layered multicast data transfer. In *Proceedings of the Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '98)*, volume 3, pages 996–1003, San Francisco, CA, USA, March 1998.

[Wat90] Andrew B. Watson. Perceptual-components architecture for digital video. *Journal of the Optical Society of America*, A 7(10):1943–1954, 1990.

[WBB04] Jörg Widmer, Catherine Boutremans, and Jean-Yves Le Boudec. End-to-end congestion control for TCP-friendly flows with variable packet size. *ACM SIGCOMM Computer Communication Review*, 34(2):137–151, April 2004.

[WDM01] Jörg Widmer, Robert Denda, and Martin Mauve. A survey on TCP-friendly congestion control. *IEEE Network Magazin*, 15(3):16–26, May 2001.

[WF01] Jörg Widmer and Thomas Fuhrmann. Extremum feedback for very large multicast groups. In *Proceedings of the Third International Workshop on Networked Group Communication (NGC 2001)*, pages 56–75, London, UK, November 2001.

[WFLG00] Thomas Wiegand, Niko Färber, Michael Link, and Bernd Girod. Analysis of video transmission over lossy channels. *IEEE Journal on Selected Areas in Communications (JSAC)*, 18(6):1012–1032, June 2000.

[WH01] Jörg Widmer and Mark Handley. Extending equation-based congestion control to multicast applications. In *Proceedings of ACM SIGCOMM 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 275–286, San Diego, CA, USA, August 2001.

[Win99] Stefan Winkler. Issues in vision modeling for perceptual video quality assessment. *Signal Processing*, 78(2):231–252, October 1999.

[WLZ01] Feng Wu, Shipeng Li, and Ya-Qin Zhang. DCT-prediction based progressive fine granularity scalable coding. In *Proceedings of the IEEE International Conference on Image Processing (ICIP) 2000*, volume 3, pages 556–559, Vancouver, BC, Canada, September 2001.

[WOZ02] Yao Wang, Jörn Osermann, and Ya-Qin Zhang. *Video Processing and Communications*. Prentice Hall, 2002. ISBN 0-13-017547-1.

[WPD88] David Waitzman, Craig Partridge, and Steve Deering. Distance vector multicast routing protocol. Internet Engineering Task Force (IETF), RFC 1075, November 1988.

[WS98] Huayang Amy Wang and Mischa Schwartz. Achieving bounded fairness for multicast and TCP traffic in the internet. In *Proceedings of the ACM SIGCOMM '98 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 81–92, Vancouver, British Columbia, Canada, September 1998.

[WZ99] Ralph Wittmann and Martina Zitterbart. *Multicast: Protokolle und Anwendungen*. dpunkt Verlag, 1999. ISBN 3-920993-40-3.

[XCRK03] Dongyan Xu, Heung-Keung Chai, Catherine Rosenberg, and Sunil Kulkarni. Analysis of a hybrid architecture for cost-effective streaming media distribution. In *Proceedings of the SPIE/ACM Conference on Multimedia Computing and Networking (MMCN 2003)*, Santa Clara, CA, USA, January 2003.

[XJ04] Dongyan Xu and Xuxian Jiang. Towards an integrated multimedia service hosting overlay. In *Proceedings of the Twelfth ACM International Conference on Multimedia*, page (to be published), New York, NY, USA, October 2004.

[YJ04] Homayoun Yousefi'zadeh and Hamid Jafarkhani. Achieving inter-receiver fairness utilizing layered media multicast control (LMMC). In *Proceedings of the IEEE International Conference on Communications (ICC 2004), to appear*, pages 1151–1155, Paris, France, June 2004.

[YJH05] Homayoun Yousefi'zadeh, Hamid Jafarkhani, and Amir Habibi. Layered media multicast control (LMMC): Rate allocation and partitioning. *IEEE/ACM Transactions on Networking (to appear)*, June 2005.

[YKL00] Yang Richard Yang, Min Sik Kim, and Simon S. Lam. Optimal partitioning of multicast receivers. In *Proceedings of the 8th IEEE International Conference on Network Protocols (ICNP 2000)*, pages 129–140, Osaka, Japan, November 2000.

[YKL01] Yang R. Yang, Min S. Kim, and Simon S. Lam. Transient behaviors of TCP-friendly congestion control protocols. In *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2001)*, volume 3, pages 1716–1725, Anchorage, AK, USA, April 2001.

[Zin03] Michael Zink. *Scalable Internet Video-on-Demand Systems*. PhD thesis, Darmstadt University of Technology, Darmstadt, Germany, September 2003.

Acronyms

3GPP	Third Generation Partnership Project
ACK	Acknowledgment
AIMD	Additive Increase Multiplicative Decrease
AQM	Active Queue Management
ARCF	Adaptive Rate Control Framework
ASM	Any-Source Multicast
CBT	Core Based Tree
CDN	Content Delivery Network
CI	Confidence Interval
DCCP	Datagram Congestion Control Protocol
DCT	Discrete Cosine Transform
DVMRP	Distance Vector Multicast Routing Protocol
DSL	Digital Subscriber Line
ECN	Explicit Congestion Notification
ELM	Equation-Based Layered Multicast
EWMA	Exponentially Weighted Moving Average
FGS	Fine Granularity Scalability

ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IP	Internet Protocol
ITU	International Telecommunications Union
KS	Kolmogorov-Smirnov
LAP	Loss Aggregation Period
LIP	Loss Insensitive Period
LWS	Light-Weight Session
MBone	Multicast Backbone
MIMD	Multiplicative Increase Multiplicative Decrease
MLE	Maximum-Likelihood-Estimator
MOS	Mean Opinion Score
MOSPF	Multicast Open Shortest Path First
MPEG	Moving Picture Expert Group
MSE	Mean Square Error
MTU	Maximum Transmission Unit
NACK	Negative Acknowledgment
PFGS	Progressive Fine Granularity Scalability
PIM	Protocol Independent Multicast
PSNR	Peak Signal-to-Noise Ratio
QoS	Quality of Service
RED	Random Early Detection

RLC	Receiver-Driven Layered Congestion Control
RMT	Reliable Multicast Transport
RTCP	Real-Time Transmission Control Protocol
RTP	Real-Time Transmission Protocol
SACK	Selective Acknowledgment
SMCC	Smooth Multi-rate Multicast Congestion Control
SNR	Signal-to-Noise Ratio
SSM	Source-Specific Multicast
TFMCC	TCP-Friendly Multicast Congestion Control
TFRC	TCP-Friendly Rate Control
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VoD	Video-on-Demand
VQEG	Video Quality Experts Group
WALI	Weighted Average Loss Interval
WWW	World Wide Web

Nomenclature

α	Significance level.
$\bar{T}_{ELM}$	Average throughput of an ELM receiver.
$\bar{T}_{RLC}$	Average throughput of an RLC receiver.
$\bar{T}_{TCP}$	Average throughput of a TCP flow.
$\beta(\cdot)$	Beta function.
χ	Estimated fair share according to the TCP throughput model.
δ	Join/leave normalized rate distance function.
η_q	Normalization factor for the quality metrics q
γ	Loss correlation factor.
$\hat{\tau}$	Maximum join/leave timer.
κ	EWMA weight parameter.
$\mathbf{g}_{l,m}$	l-Tuple of ordered group rates $g_l, \ldots, g_{l+m-1}$.
$\mathcal{G}_{l,m}$	Set of all possible m-tuples of receiver group rates $g_l, \ldots, g_{l+m-1}$.
μ	Mean of a Gaussian (normal) distribution.
$\nu_1, \ldots, \nu_3$	Parameters of the FGS square-root model.
Ψ_i	Impact of the ith loss event.
Ψ_{avg}	Average loss impact.

σ	Variance of a Gaussian (normal) distribution.
τ	Join/leave timer.
τ_{deaf}	Deaf period.
$\tau_{start-up}$	Time between to joins in the start-up phase.
Θ_0	Current non-terminated loss interval.
Θ_{avg}	Average loss interval.
Θ_i	The ith most recent and terminated loss interval.
ξ_i	Acceptable loss tolerance of receiver i.
ζ	Join/leave Sensitivity function.
$B(\cdot)$	Beta distribution.
$BD(\cdot)$	Binomial distribution.
C	Link capacity.
c_i	Bandwidth capability of receiver i.
$D(R)$	Rate-distortion function.
E	Expected value.
E	Expected value.
F	Aggregate utility fairness.
F_1	Aggregate utility fairness of a single receiver group.
F_{ELM}	TCP-fairness index of ELM.
f_i	Fairness index of receiver i.
F_{RLC}	TCP-fairness index of RLC.
g_l	Transmission rate of the lth receiver group.
L	Number of receiver groups in a session.

l	Quality level, receiver group, subscription level.
$L(\cdot)$	Likelihood function.
N	Number of participating receivers in a multicast session.
$N(\cdot)$	Normal distribution.
n^*	Targeted number of feedback messages.
n_{ack}	Number of packets acknowledged by a single TCP ACK.
n_{lost}	Number of lost packets.
$n_{lost}[i]$	Number of lost packets within the ith loss interval.
n_{RTT}	Number of packets sent within a round-trip time.
$n_{transmitted}$	Total number of transmitted packets.
p	Probability.
p_{drop}	Packet drop probability.
p_{event}	Loss event rate.
$p_{fraction}$	Loss fraction.
$pdf(\cdot)$	Probability density function.
q_l	Quality of the video signal transmitted to the lth receiver group.
r_i^*	Optimal operational rate or fair share of receiver i.
r_i	Actual operational rate of receiver i.
r_{base}	Minimum data rate supported by the encoded video.
r_{enh}	Enhancement layer bit rate.
r_{max}	Maximum available data rate of the encoded video.
S	Reference packet size.
s	Actual packet size.

T	Throughput.
t_{RTO}	Retransmission timeout.
t_{RTT}	Round-trip time.
U	Intra-session utility fairness.
u	Utility function.
u_i	Utility of receiver i.
V	Variance.
w_i	Weight of the ith loss interval.
X	Random variable.
$x[\cdot]$	Sequence of samples of original video frame.
x_i	ith symbol of original video frame.
x_{base}	Base layer symbol of original video frame.
x_{enh}	Enhancement layer symbol of original video frame.
x_{FD}	Symbol of original video frame in the frequency domain.
x_{PD}	Symbol of original video frame in the picture domain.
x_{pp}	Peak-to-peak value of original video frame.
$y[\cdot]$	Sequence of samples of reconstructed video frame.
y_i	ith symbol of reconstructed video frame.
y_{base}	Base layer symbol of reconstructed video frame.
y_{enh}	Enhancement layer symbol of reconstructed video frame.
y_{FD}	Symbol of reconstructed video frame in the frequency domain.
y_{PD}	Symbol of reconstructed video frame in the picture domain.
z	z-value of the unit normal distribution.