BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS
FACULTY OF ELECTRICAL ENGINEERING AND INFORMATICS
DOCTORAL SCHOOL OF INFORMATICS
DEPT. OF TELECOMMUNICATIONS AND MEDIA INFORMATICS

ANALYSIS AND DESIGN OF RADIO ACCESS TRANSPORT NETWORK CONGESTION CONTROL AND DIMENSIONING METHODS

Pál L. Pályi

MSc. in Technical Informatics

Ph.D. Dissertation

Supervised by

Dr. Sándor Molnár
High Speed Networks Laboratory,
Dept. of Telecommunications and Media Informatics

Advisors:
Dr. Sándor Rácz* and Dr. József Bíró‡
*Ericsson Research, Budapest; ‡HSN Lab.

Budapest, Hungary
2013

Abstract

The increasing number of mobile broadband users and their even more increasing data demands require the mobile telecommunications networks to be adapted to these changes rapidly. The elaboration of several new congestion avoidance methods are needed in these networks to ensure the appropriate Quality of Service (QoS) in the rapidly saturating network environment. Since the radio part of the network is often enhanced faster than the terrestrial part due to cost-efficiency reasons, the terrestrial part may be a potential bottleneck for the whole network. That is why the present dissertation focuses on this part of the network and addresses fairness issues, presents efficient congestion control methods and solves open issues in dimensioning methods that can be applied in the Transport Network (TN) of Radio Access Networks (RAN) in the 3rd Generation Partnership Project (3GPP) High-Speed Packet Access (HSPA) system.

Fairness of resource sharing is an important issue in all systems and networks. The importance is even higher if the resources are limited. The present dissertation deals with fairness of sharing the limited resources in the RAN Transport Network. A fairness-optimal flow initialization method is proposed that can be used in a rate- or window-based congestion control where flows share the same bottleneck. The PhD thesis also gives a general solution for choosing fairness-optimal initial rate in the case of second-order fairness measures.

Since the RAN TN may be a potential bottleneck, congestion control is needed on the transport network. The dissertation proposes a novel, non-standardized, cross-layer, window-based HSDPA (High-Speed Downlink Packet Access) Transport Network Congestion Control. This solution uses the standardized congestion detection and signalling framework and it is based on extending the Radio Link Control (RLC) protocol with congestion control functionality.

In order to decrease the chance of RAN TN congestion occurrences in the long run, besides congestion control, link dimensioning is an important issue, as well. For link dimensioning purposes rate sharing models are needed. The dissertation characterizes the state space of the Discriminatory Processor Sharing (DPS) service discipline with peak-rate limitations of the flows as an appropriate rate sharing model. An efficient algorithmic approach is presented to determine which classes are subject to peak-rate limitations and based on this the bandwidth shares of flows of classes in a given state of this system.

Finally, in the dissertation another bandwidth sharing model, the integration of stream and elastic traffic is analyzed on flow level; the average throughput is calculated for elastic traffic classes. The stationary probabilities of macro states are characterized and the accuracy of the approximation method is evaluated.

Abbreviations

3G	3rd Generation
3GPP	3rd Generation Partnership Project
ACK	Acknowledgement
ARQ	Automatic Repeat Request
ATM	Asynchronous Transfer Mode
CAC	Call Admission Control
CC	Congestion Control
DCH	Dedicated Channel
DL	Downlink
DRT	Delay Reference Time
E-DCH	Enhanced DCH
GoS	Grade of Service
GSM	Global System for Mobile Communications
HS	High Speed
HS-DSCH	High Speed Downlink Shared Channel
HSDPA	High Speed Downlink Packet Access
HSPA	High Speed Packet Access
IP	Internet Protocol
LTE	Long Term Evolution
MAC	Medium Access Control
PDU	Protocol Data Unit
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
RAN	Radio Access Network
RLC	Radio Link Control
RNC	Radio Network Controller
RSD	Relative Standard Deviation
RTT	Round Trip Time
TCP	Transmission Control Protocol
TN	Transport Network
TTI	Transmission Time Interval
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
UTRAN	UMTS Terrestrial RAN
WCDMA	Wideband Code Division Multiple Access

Acknowledgements

First of all, I would like to say thanks to all of my three advisors for their support. Especially to my industrial advisor, Dr. Sándor Rácz at Ericsson Research Hungary who introduced me as still an MSc student to the very interesting research field of mobile broadband and guided me throughout all my research activities. His exceptional professional knowledge, insight, enthusiasm, encouragement and support have given me an essential factor of motivation to finish (and even start) the present PhD thesis.

Most of the work presented in this dissertation was completed at the Traffic Analysis and Network Performance Laboratory of Ericsson Research Hungary. I wish to say thanks to my colleagues and former colleagues for their support during the work; especially Szilveszter Nádas, Zoltán Nagy, István Ketykó and Balázs Péter Gerő.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Modern mobile telecommunication networks and high speed data packet services (like 3G/HSPA/LTE, see 3GPP (2008d), Dahlman et al. (2008)) need the elaboration of several new congestion avoidance methods to ensure the appropriate Quality of Service (QoS). These methods are needed because the radio part of the network is often enhanced faster than the terrestrial part, making the terrestrial part a potential bottleneck for the system. During the development of the methods, it is not enough to apply existing techniques, but the characteristics of the network under development and the quality demanded by future users should be taken into account as well. Throughout this adaptation, many challenging assignments are to be solved in the scientific point of view.

1.1 Scope and motivation of research

The present thesis considers the Transport Network (TN) of the Radio Access Network (RAN) in the 3rd Generation Partnership Project (3GPP) High Speed Packet Access (HSPA) and Long Term Evolution (LTE) (Figure 1.4, pp. 7) systems. This network may be a potential bottleneck in many cases. Moreover, service differentiation is needed in a modern network for various reasons. In order to serve these new demands, elaboration of new methods are needed.

1.1.1 RAN Transport Network as a bottleneck

The RAN Transport Network may be a potential bottleneck for the whole system in many scenarios, since it is often challenging to provide the appropriate capacity (availability) of the network and control the cost at the same time, see Nadiv & Rosenhouse (2010). Moreover, the radio part of the network is often enhanced faster than the terrestrial part, which also makes the terrestrial part a potential bottleneck. This "cost challenge" means basically to keep the Total Cost of Ownership (TCO)

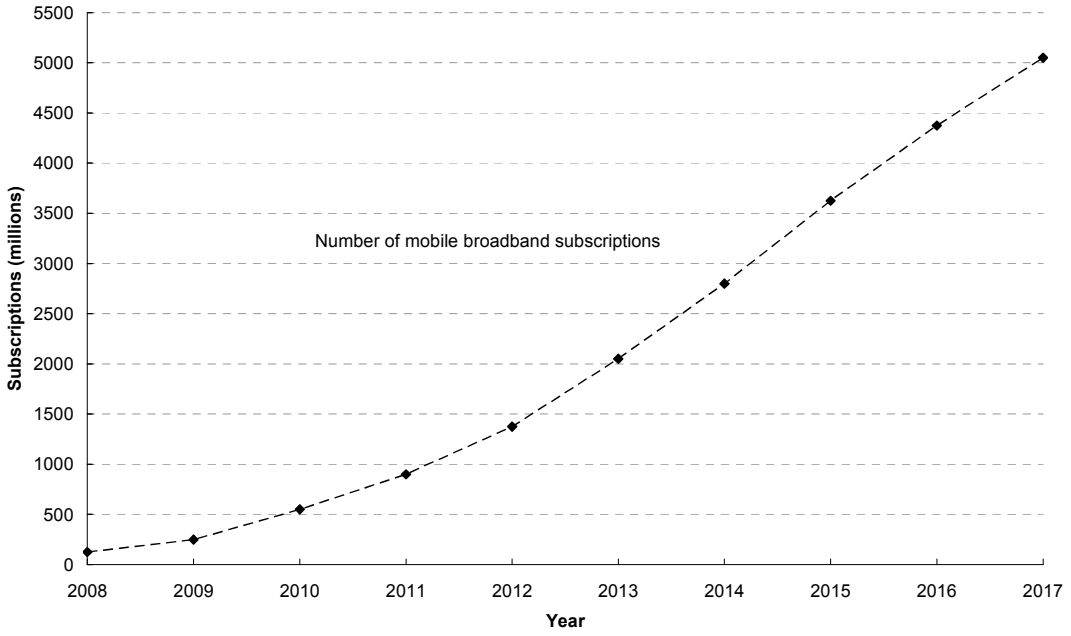

Figure 1.1: Mobile broadband subscriptions, 2008-2017 (Ericsson (2012))

of a network low when migrating to higher broadband capacities. It is essential to handle the challenge of capacity and cost in the mobile broadband backhaul networks where data traffic is dramatically increased.

Data demands of today had not been foreseen when operators made investments in the High-Speed Packet Access (HSPA) technology, even just a few years ago (see Ericsson White Paper (2011b)). Recent Cisco White Paper (2012) states that global mobile data traffic grew almost threefold in 2010, nearly tripling for the third year in a row. According to Cisco White Paper (2012), global mobile data traffic will increase 26-fold between 2010 and 2015. According to a very recent report of Ericsson (2012), mobile data traffic almost doubled between the first quarter of 2011 and 2012. Moreover, the number of mobile broadband subscriptions already reached 1 billion, and will reach 5 billions in 2017 (Ericsson (2012)), see Figure 1.1.

As a consequence, capacity and cost are the key factors in making decisions in planning backhaul networks. Using the existing infrastructure efficiently, in combination with a migration towards IP/Ethernet architecture and the right level of network automation, plus effective operation, administration and maintenance (OAM) tools will decrease the Total Cost of Ownership (TCO), see Ericsson White Paper (2008).

In non-urban areas, installing optical fiber cables in the transport network is not cost-efficient according to Ekstrom et al. (2006), Ericsson (2012). Therefore, alternative technologies – providing typically lower bandwidth – are used.

Microwave transmission often provides the lowest TCO in cases where no infrastructure is available at the base station site, and it effectively scales in capacity.

With the evolution to mobile broadband, operators tend to invest in self-owned microwave networks to keep their cost at check, see Ericsson White Paper (2008). Apart from rural areas, microwave is frequently used in the case of new roll-out of mobile backhaul networks, where the amount of the expected traffic is still not known and therefore no fixed solutions are used. Consequently, one of the most widely spread alternative technology is the microwave point-to-point transmission (Ericsson White Paper (2008)). Recent report by Metro Ethernet Forum (2011) states that Wireless point-to-point microwave solutions are used in over 60% of all mobile backhaul deployments worldwide. These solutions offer operators a simple, cost-efficient way for upgrading and expanding their networks. They support higher data rates than copper T1/E1 lines, and easily overcome the high cost and limited availability associated with fiber, see Figure 1.2. In this figure, 1+1 refers to a configuration where a radio connection is composed of an active link and a secondary backup link, whereas 1+0 refers to a basic transmission setup without any redundancy level. As it can be seen in Figure 1.2, the cost of microwave links – unlike fiber links – is independent of the distance. Moreover, the latest microwave solutions by Ericsson (2011) may reach air interface traffic capacity up to 406 Mbps (using 512 Quadrature Amplitude Modulation (QAM) at 56 Mhz). However, microwave backhaul has its constraints; spectrum is a scarce resource, therefore, microwave transmission based transport networks may still represent a bottleneck (Ericsson White Paper (2008)).

Besides microwave transmission, leased lines or shared transport is also used as a low-cost alternative of optical fiber cable based transport network, typically in the USA. In North America, the most frequently chosen backhaul technologies are copper or fiber lines because the availability of the wireless spectrum for microwave links was very limited when the existing mobile networks were being built. For example, one major operator used 50 percent fiber, 15 percent microwave links and 35 percent leased copper lines according to Ericsson White Paper (2008).

In these cases the Transport Network may represent a bottleneck. The Transport Network using optical fiber cables may also be a bottleneck in the case of a common HSPA and LTE transport, where more NodeBs' and eNodeBs' traffic are multiplexed together. If an operator that provides HSPA services introduces the LTE system, the existing 3G/HSPA architecture may be reused for cost-efficiency reasons, see Ekstrom et al. (2006). Most of the new eNodeBs will probably be connected to the existing Transport Network if possible, no new transport infrastructure will be deployed, so HSPA and LTE will probably have a common Transport Network, which can be again a potential bottleneck because of the increasing Uu peak rates.

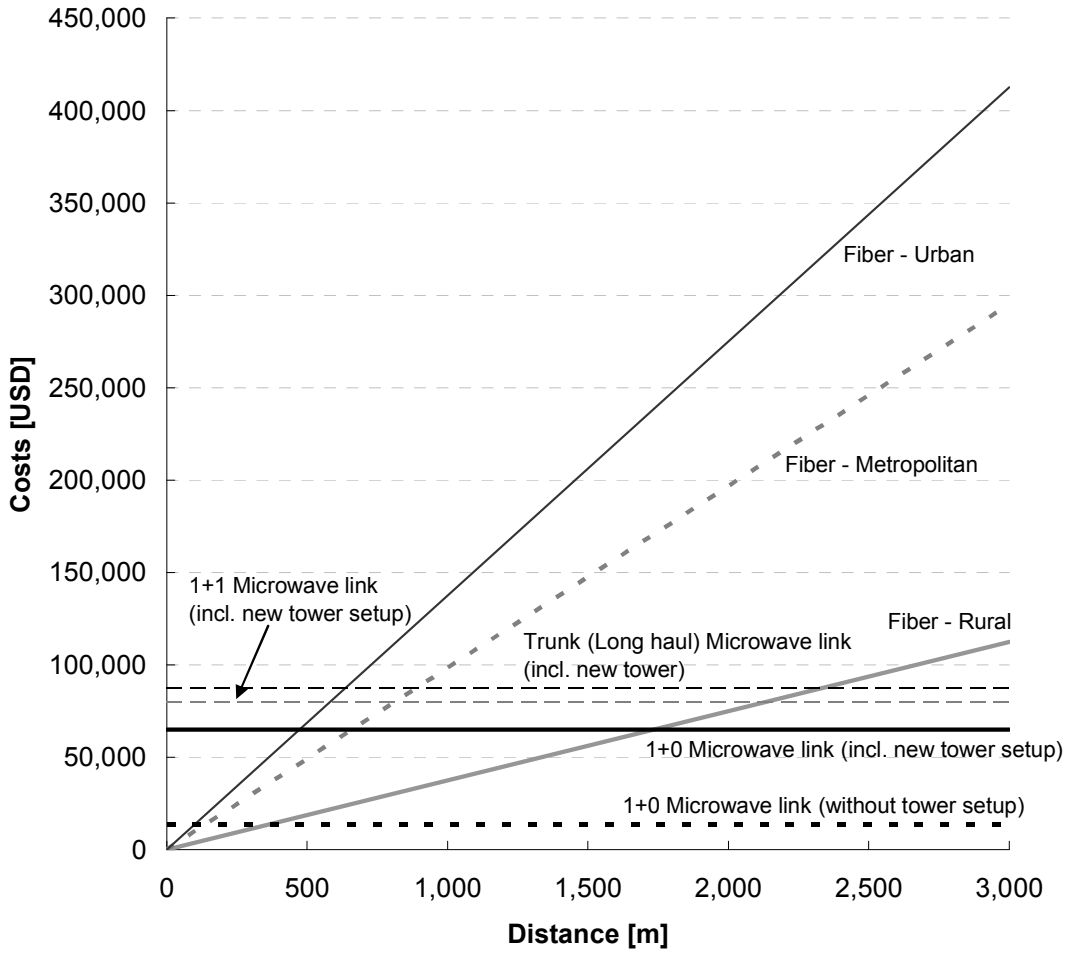

Figure 1.2: Microwave vs. fiber Joseph & Avital (2010)

1.1.2 Why service-differentiation in RAN is needed

Service differentiation is necessary to support the business model used by the operator (see Cisco White Paper (2012), Sprenkels et al. (2000)) or to efficiently handle the increasing traffic, so service differentiation is basically needed due to at least two reasons.

The first one is a marketing and pricing related. Operators often offer different service packages for different prices, e.g., 'gold', 'silver', 'bronze' subscriptions (see Garriga et al. (2009), Sprenkels et al. (2000)), where 'gold' represents the best service (e.g., highest bandwidth or priority), and 'bronze' the worst. This should be supported by the transport network as well, see Ekstrom et al. (2006). This marketing related service differentiation needs technology support. See Figure 1.3 for an example. In both side of this figure *aggregated* flow rate vs. time is plotted. Aggregated means that the dark gray line represents the sum of rates of Flow 1 and Flow 2, and the black line (which equals to the capacity) represents the sum of rates of Flow 1, Flow 2 and Flow 3. On the left side of the figure Flow 1 and Flow 2 belongs to the 'bronze' service class whereas Flow 3 belongs to the 'gold' service class. Therefore,

the rate of Flow 3 is relatively twice as high as that of Flow 1 and Flow 2. On the right side of Figure 1.3, all the three flows belong to the 'bronze' class. Therefore they have equal rates.

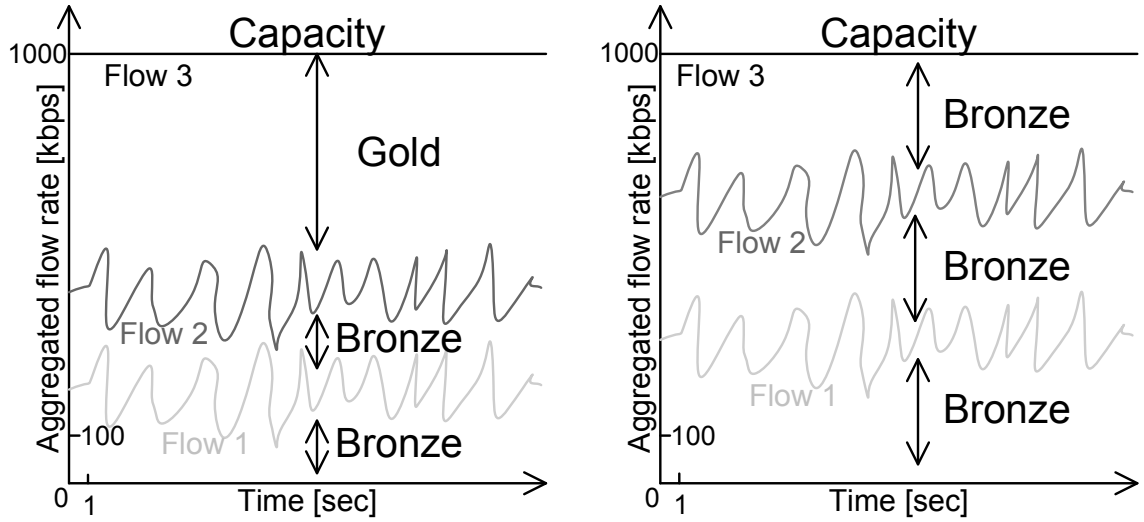

Figure 1.3: Service differentiation

The second reason is the handling of heavy users that typically fall into this category due to peer-to-peer file sharing. Report by Schulze & Mochalski (2009) states that Peer-to-peer file sharing (P2P) still generates by far the most traffic ranging from 43 percent in Northern Africa to 70 percent in Eastern Europe. This traffic is relevant for mobile access too according to Heikkinen et al. (2009), so it has its impact on HSPA/LTE traffic as well. These users represent a threat to network stability and they are one of the causes of the 'revenue gap', also known as the scissor effect, see He & Walrand (2006). There are various solutions for managing the relatively small number of heavy users who generate most of peer to peer traffic. E.g., banning the heavy user, limiting its access for a period of time, or handle it adaptively in a dynamic way like assigning it to the 'bronze' class, see Ericsson White Paper (2011a). The most effective and forward-looking one is to assign the heavy user to the aforementioned bronze class after he/she has reached the limit, e.g., daily 1 GB. In this way the heavy user cannot starve other users in the busy hour but can download freely in underutilized situations, e.g., at night. It is a good solution even if the user pays more for 'unlimited' access.

In the Radio Access Network, the bottleneck is either the radio interface or the Transport Network. Service differentiation is only meaningful in the case of a bottleneck. On one hand, the radio interface as a bottleneck is handled sophistically by existing solutions, e.g., scheduling. On the other hand, Transport Network bottleneck handling still has its research challenges. In order to make good service differentiation, both parts should be taken special care of and the methods on both parts should be

well aligned with each other. A good example is given by Garriga et al. (2009); they present the results of a QoS Load Differentiation trial performed in a real network environment using a mix of real life mobile broadband applications.

1.2 Technical background

In this section we give an overview of the architecture and technology that is needed to understand the background and the motivation of the theses.

The Radio Access Network (RAN) Transport Network (TN) of a 3G or 3GPP Long-Term Evolution (LTE) system comes with new problems to be solved. RAN TN may be a bottleneck in the case of high speed data packet services therefore it needs special handling in terms of congestion avoidance. This thesis considers the Transport Network of the 3G High Speed Downlink Packet Access (HSDPA) system and the proposed methods methods are adapted to this system.

1.2.1 Radio Access Networks

The present thesis deals with two special kind of RANs; 3G/HSPA and LTE. For other purposes other types of RANs are also used.

Radio Access Networks (RAN) are used to provide connection between a mobile terminal and the core network. The main purpose of RANs is to isolate all the radio issues from the core network, see Dahlman et al. (2008). RANs can be grouped by the radio access technology they implement. Radio access technologies are developed by various standardization institutions such as IEEE or the 3rd Generation Partnership Project (3GPP).

IEEE continuously enhances the IEEE 802.11 Wireless LAN (WLAN) standard (often referred to as Wireless Fidelity, WiFi) to provide higher and higher data rates. The IEEE 802.16, the World-wide Interoperability for Microwave Access (WiMax), has been recently standardized and considered as the metropolitan area extension of WiFi. Presently, a new Orthogonal Frequency Division Multiplexing (OFDM) based cellular standard is under preparation as a less sophisticated but more cost-effective alternative of the 3G systems of 3GPP.

The main scope of the present thesis are RANs that implement cellular radio access technologies by 3GPP, e.g., Global System for Mobile Communications (GSM), Wideband Code Division Multiple Access (WCDMA), High-Speed Packet Access (HSPA) and Long Term Evolution (LTE).

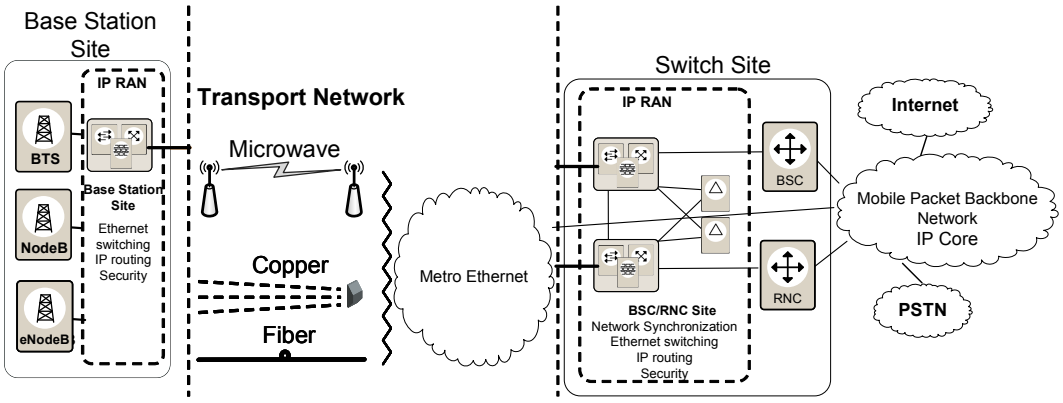

Figure 1.4: RAN Transport Network in a mobile telecommunications system

1.2.2 WCDMA RAN Transport Network

The RAN Transport Network connects the Radio Base Station (RBS) Site and the Switch Site and may use multiple physical link technologies, e.g., microwave, copper and fiber, see Ericsson White Paper (2008). Figure 1.4 shows the place of the RAN Transport Network in a GSM/3G/HSPA/LTE system.

The WCDMA RAN handles all tasks that relate to radio access control, such as radio resource management and handover control. The core network, which is the backbone of WCDMA, connects the access network to external networks (Public Switched Telephone Network (PSTN), Internet). The user equipment (Mobile Terminal or Mobile Station) is connected to radio base stations (Node B) over the WCDMA air interface (Uu). During soft handover, one User Equipment (UE) can communicate with several Node Bs simultaneously.

With the introduction of HSDPA, the RAN Transport Network may already be a bottleneck for the system. The Transport Network can also be congested because for HSDPA flows no resource reservation is made. In practice, the increased air interface capacity of HSDPA did not always come with similarly increased Transport Network capacity. Network operators often upgrade the base stations (Node Bs) first and delay the upgrade of the Transport Network until there is significant HSDPA traffic. In some cases also the cost of Iub transport links is still high, however it decreases significantly at many places with the introduction of new mobile backhaul technologies, see Ericsson White Paper (2008). Thus it is a common scenario that the throughput is limited by the capacity available on the Iub Transport Network links and not by the capacity of the air interface.

These issues provoke new research challenges. The RAN Transport Network needs special handling in terms of congestion avoidance. Figure 1.5 shows techniques for avoiding congestion on different time scales, see Jain (1995). For networks that are almost always congested, installing higher capacity links and redesigning the topology

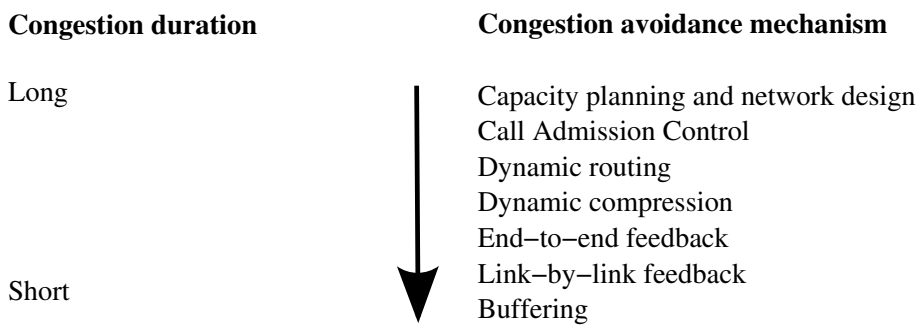

Figure 1.5: Congestion duration vs. congestion avoidance techniques (Jain (1995))

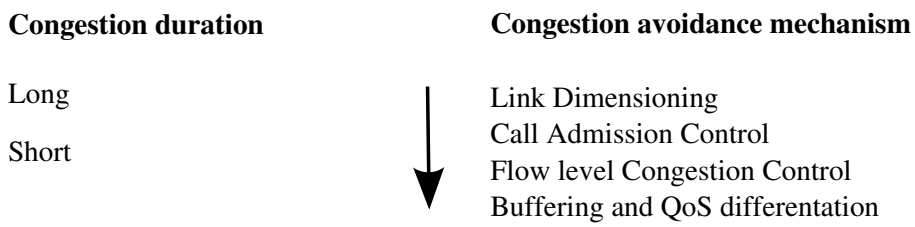

Figure 1.6: Congestion duration vs. congestion avoidance mechanism

is the best solution.

If congestion occurs sporadically one method is to route according to the load level of links and to apply Connection Admission Control. For congestions lasting less than the connection duration – but not very short – an end-to-end control scheme can be used. For very short spikes in traffic load, applying adequate buffers in the switches is the best method.

Figure 1.6 depicts the congestion avoidance mechanisms proposed to be used in the WCDMA RAN transport network. For very short data burst the most appropriate solution is to provide sufficient buffers and Quality of Service (QoS) differentiation. This provides small delay for high priority traffic, while buffering of low priority, less delay sensitive traffic increases utilization. For traffic with no fixed bandwidth requirement, congestion control can be used to adapt the bitrate to the actually available resources. For guaranteed bitrate (GBR) services Call Admission Control (CAC) can be used to check whether there is enough free capacity in the system. With adequate link dimensioning the desired Grade of Service (GoS) can be guaranteed for a given traffic model.

The scope of the present dissertation is congestion control and dimensioning. The scarce resources of the RAN Transport Network should be carefully dimensioned. Moreover, a system-specific congestion control is needed because the Radio Link Control (RLC) protocol (specified by 3GPP (2009)) does not have congestion control functionality and the TCP congestion control can not operate efficiently above the Acknowledged Mode RLC.

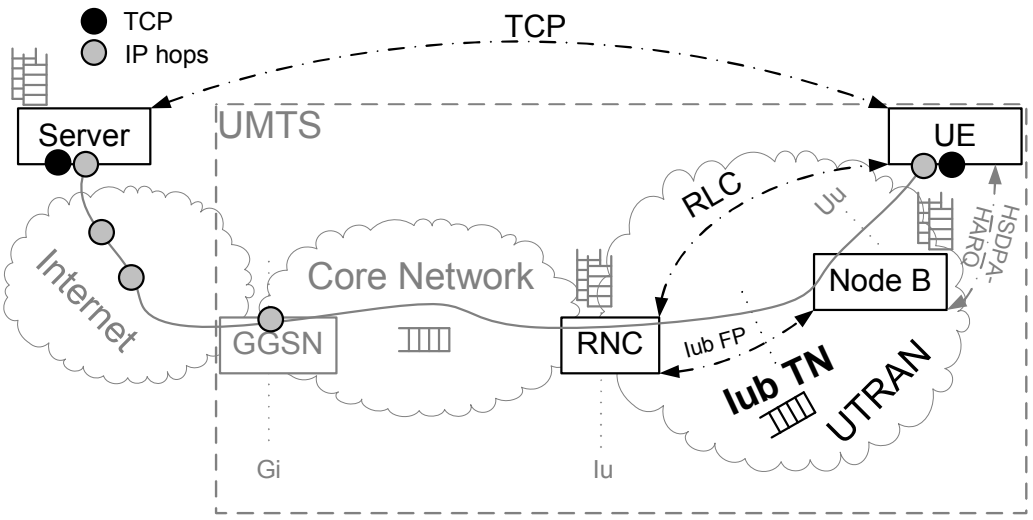

Figure 1.7: Control loops in 3G/HSDPA

1.2.3 HSDPA RAN Transport Network Architecture

The introduction of High Speed Packet Access (HSPA) greatly improves achievable bitrate but it presents new challenges to be solved in the Wideband Code Division Multiple Access (WCDMA) radio access network (RAN). For Dedicated Channels (DCHs), transport network bandwidth can be reserved by means of admission control. Bandwidth reservation is not efficient for HSDPA, because of the higher peak rates and much higher variance of achieved bitrate thus a new solution is needed to control congestion. In the Internet such congestion is controlled by e.g. the end-user TCP, but that is not possible in the RAN transport network, because lost packets are retransmitted by RLC.

Potentially, several control loops can be found in the downlink to resolve congestion (as shown in Figure 1.7). The main architectural difference between HSDPA and Release 99 regarding the loops in the system is that HARQ is only present in the case of HSDPA, since in Release 99 originally RLC handles radio link failures.

In response to the increased need for higher bitrate and more efficient transmission of packet data over cellular networks, the WCDMA 3GPP Release 5 extended the WCDMA specification with the High Speed Downlink Packet Access (HSDPA), see 3GPP (2004). The demand for uplink performance improvement is addressed by introducing Enhanced Dedicated Channel (E-DCH) – often referred as Enhanced Uplink (EUL) or High Speed Uplink Packet Access (HSUPA) – in 3GPP Release 6, see Parkvall et al. (2005). HSDPA and HSUPA together are called HSPA. The main architectural novelty of HSPA is that certain parts of the control of radio resources have been moved from RNC to Node Bs. In 3GPP Release 7 higher-order modulation and multiple input multiple output (MIMO) are introduced for HSDPA to further improve the achievable bitrate, see Bergman et al. (2009).

For HSDPA a new shared downlink transport channel, called High-Speed Downlink Shared Channel (HS-DSCH), is also introduced. This channel is dynamically shared among packet data users, primarily in the time domain, the Transmission Time Interval (TTI) is 2 ms. The application of shared channel makes the use of available radio resources in WCDMA more efficient. HSDPA also supports new features that rely on the rapid adaptation of transmission parameters to instantaneous radio conditions. The main principles are fast link adaptation, fast Hybrid-ARQ (HARQ) with soft-combining and fast channel-dependent scheduling, see Parkvall et al. (2003). In order to support HSDPA with a minimum impact on the existing radio interface protocol architecture, a new Medium Access Control (MAC) sub-layer, MAC-hs, has been introduced for HS-DSCH transmission which is implemented in the Node B and the User Equipment. In the Node B, the radio scheduler entity is responsible for distributing resources of the air-interface among Priority Queues (PQ) – which are buffers in the Node B – per Transmission Time Interval. In the given TTI, maximum a Transport Block Size (TBS) amount of data is scheduled and forwarded from the chosen PQ to the appropriate HARQ process towards the User Equipment.

1.2.3.1 Transport Network Congestion Control solutions

The issue of HSDPA Transport Network Congestion Control has already been addressed in the literature. Nádas et al. (2007) proposed a rate-based Congestion Control that operates in the Iub Framing Protocol (FP) layer. It performs well independently of the Transport Network topology. Bajzik et al. (2006) introduced a cross-layer (protocol design by the violation of a reference layered communication architecture, see Srivastava & Motani (2005)) backpressure algorithm located in the RNC that is pro-actively preventing congestions and RLC retransmissions. Their method is based on controlling the ATM/AAL2 buffer of the outgoing link at RNC. That is not possible, when the bottleneck buffer is in the Transport Network, but not in the RNC or in the case of a general TN. Weigle et al. (2005) presented a new congestion detection and reaction mechanism for TCP, based on measurements of One-way Transit Times (OTT) of TCP segments within a TCP connection. OTTs can more accurately estimate the forward-path queuing delay than round-trip times (RTT). If the RTT is increasing, the sender can not find out whether it is caused by congestion on the forward (data) path, congestion on the reverse (ACK) path, or both.

1.2.3.2 Rate-based per-flow Transport Network Congestion Control

In this section we review the Congestion Control (CC) solution proposed by Lundh et al. (2008). This solution is extended in Chapter 2 and considered in Chapter 3.

The HSDPA Transport Network Congestion Control by Lundh et al. (2008) operates in a per-flow basis, i.e., each HSDPA flow has its own congestion detection, bitrate calculation and shaper part (see Figure 1.8). The main tasks of these three parts are as follows.

The *congestion detection part* in the base station (Node B), aims at finding out the congestion level of the Transport Network based on packets arriving from the Radio Network Controller (RNC) and on the state of the Node B buffers (referred to as MAC-hs Priority Queue in 3GPP (2008a)). Standard 3GPP congestion detection functionalities are used like by Nádas et al. (2007). If Transport Network congestion is detected the bitrate calculation part will be informed.

The *bitrate calculation part* in the Node B calculates the bitrate that is allowed on the Transport Network for the given flow. The congestion control maintains an internal variable for the maximum bitrate of the flow. This bitrate is increased linearly if there is no Transport Network congestion (no reported congestion from congestion detection part). If congestion is reported, the bitrate is reduced multiplicatively. The reduction rate depends on the type of congestion. When a new flow arrives – and therefore a new congestion control entity is created – a Slow Start-like mechanism is used to find out the proper starting bitrate of the flow. It means that the rate is exponentially increased during each round round trip time interval until congestion occurs. After the first congestion the congestion control behaves the above described AIMD manner. The bitrate is calculated per 100 ms. This value is a compromise among fast reaction, low frequency of Capacity Allocation (CA) control frame sending and low calculation complexity, see Nádas et al. (2007). If the calculated bitrate of the flow changes, then the shaper will be informed about the new bitrate through the Capacity Allocation control frame.

The *shaper* in the RNC shapes the flow according to the signaled maximum flow bitrate. This bitrate is coming from the latest received Capacity Allocation control frame.

1.3 Methodology

In the first part of the thesis, algorithm design and validation by analytical methods and simulations represent most of the work. The simulation tools were a simple self-developed flow-level Additive Increase Multiplicative Decrease (AIMD) simulator and a complex packet-level HSDPA protocol simulator. The protocol simulator was needed because the proposed method was applied in a high-complexity system that cannot be evaluated by analytical methods.

In the analysis of infocommunication systems simulation tools are frequently used

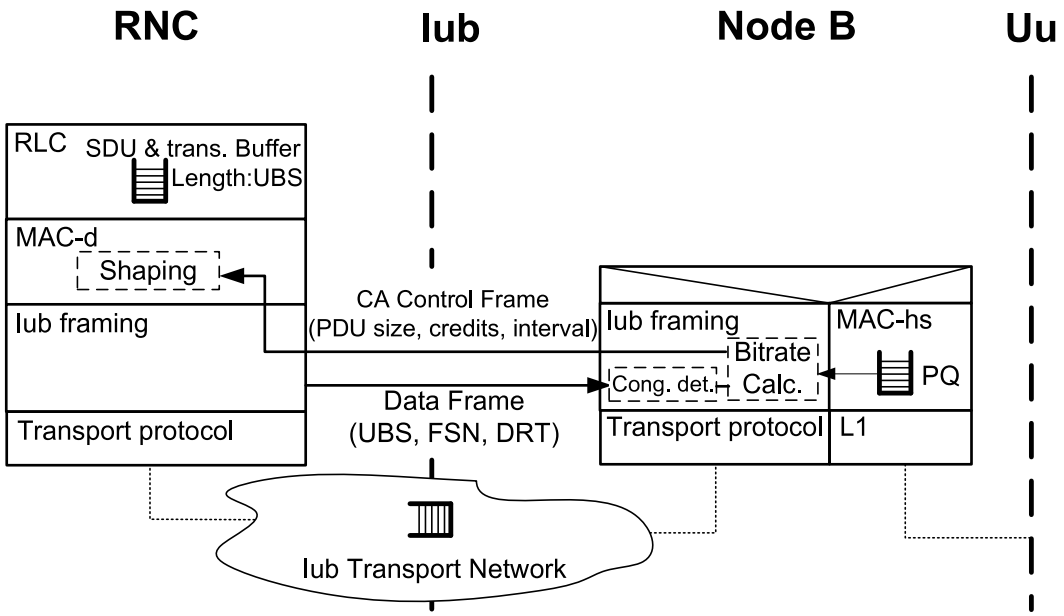

Figure 1.8: Congestion Control in UTRAN

because of the high system complexity. Complex protocol interactions and the radio interface cannot always be modeled analytically. Therefore, we make a simulation model of the system where the whole protocol stack is implemented in detail along with the radio environment which consists of standard models for distance attenuation, shadow fading and multipath fading, based on 3GPP typical urban channel model. The applied simulation tool was developed by a vendor and product development within the system is based on it. It means that before a new release of a product the new functions are implemented and evaluated in this detailed protocol simulator accurately modeling a WCDMA/HSDPA system. If simulations provide satisfying results the new function is likely to be implemented in the corresponding product also. We have always studied extensive simulation scenarios considered relevant by the vendor from the product point of view. Therefore this simulation tool can be considered a more precise model of the system than commonly used simulation tools, e.g., the well-known NS-2. Results provided by this simulator almost correspond to measurements in the modeled system.

In the second part of the thesis, numerical method development was validated by mathematical methods.

1.4 Outline of the dissertation

Fairness of resource sharing is an important issue in all systems and networks. The importance is even higher if the resources are narrow. Chapter 2 deals with fairness of sharing the limited resources in the RAN Transport Network. A fairness-optimal

flow initialization method is proposed that can be used in a rate- or window-based congestion control where flows share the same bottleneck. The chapter also gives a general solution for choosing fairness-optimal initial rate in the case of second-order fairness measures. The theoretical results are applied and studied in a rate-based RAN Transport Network congestion control for HSDPA to provide fairness-optimal initial rate for incoming HSDPA flows.

Since the RAN Transport Network may be a potential bottleneck, congestion control is needed on the transport network. In Chapter 3.1 a novel, non-standardized, cross-layer, window-based HSDPA Transport Network Congestion Control is proposed. This solution uses the standardized congestion detection and signalling framework and it is based on the idea of extending the Radio Link Control (RLC) protocol with congestion control functionality.

In order to decrease the chance of RAN Transport Network congestion occurrences in the long run, besides congestion control, link dimensioning is an important issue, as well. For link dimensioning (Jain (1995), [B1]) purposes rate sharing models are needed, especially for the elastic-type (compressible) traffic flows. In Chapter 4, we characterize the state space of the Discriminatory Processor Sharing (DPS) service discipline with peak-rate limitations of the flows. A DPS scheme is considered to model service differentiation in the network. We analyze a bandwidth-efficient rate sharing model, in which the server capacity unused by peak-rate limited flows is re-distributed among the non-limited flows. An efficient algorithmic approach is presented to determine which classes are subject to peak-rate limitations and based on this the bandwidth shares of flows of classes in a given state of this system.

In Chapter 5, another bandwidth sharing model, the integration of stream and elastic traffic – which is an adequate model for voice and data services – on flow level is analyzed; the average throughput is calculated for elastic traffic classes. A two-dimensional macro-state representation is introduced for the micro-state model. The stationary probabilities of macro states are characterized and the accuracy of the approximation method is evaluated.

Finally, the dissertation is concluded in Chapter 6.

Chapter 2

Fairness-optimal flow initialization

Fairness of resource sharing is an important issue in systems and networks, generally. If the resources to be shared are quite limited – like in the case of RAN Transport Network (TN) – fairness has even higher importance. This chapter addresses the improvement of the fairness characteristics of such a system.

The proposed method is presented as part of an HSDPA congestion control, but it can be applied in other parts of the RAN TN and in general in any rate-based congestion control where flows share the same bottleneck. The main idea behind the proposed method is improving fairness at new flow arrivals. The most widely used congestion control scheme, called Additive Increase Multiplicative Decrease (AIMD) only guarantees convergence to fairness in the long run. It means that the flow rates converge to a fair (also considering the users' subscription, see Figure 1.3) share of resources in steady state where no flows join or leave. However, incoming flows may decrease the level of fairness, that is why transient fairness should be taken into account as well. AIMD does not define the starting rate of a flow. The scheme deals only with the dynamic behaviour of flows, after congestion. Therefore, when a new flow arrives it obviously needs an initial rate to start from, see Figure 2.1. This initial rate influences fairness of resource sharing at the time of the flow arrival. Obtaining better fairness directly at flow arrivals affects the long-term fairness positively because the convergence time to fair bandwidth share will be shorter.

The method is based on finding the fairness-optimal initial rate of a new flow as a function of the rates of ongoing flows. With this fairness-optimal initial value, fairness can be improved to the greatest possible extent. This fairness-optimal solution is determined for the most widely used fairness measure, called Jain's fairness index and generally for a group of second-order fairness measures.

Research results related to this chapter are published in [C1], [C2] and [J3]. Related patent application is in [P1]. My theses are related to all the main results of this chapter.

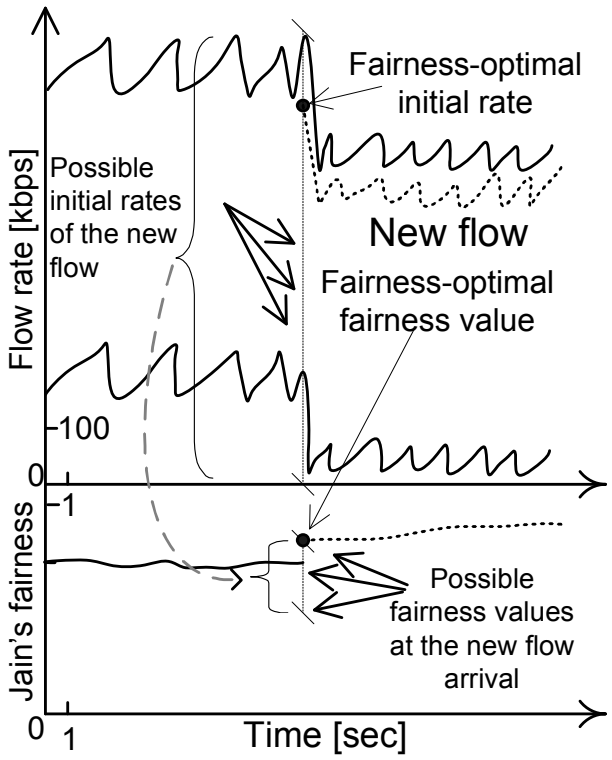

Figure 2.1: Flow initialization

2.1 Fairness measures

The notion of fairness has been defined several ways in the literature. What is considered fair depends on many factors. Mo & Walrand (2000) reviewed, compared and generalized standard definitions of fairness. One of the most common ones is Max/min or bottleneck optimality criterion. A feasible flow rate x is defined to be Max/min fair if any rate x_i cannot be increased without decreasing some x_j which is smaller than or equal to x_i. Another proposition by Chiu (2000) is the notion of proportional fairness which suggests allocating more bandwidth to shorter flows in many scenarios. Norlund et al. (2004) define new fairness measures that are better suited for measuring fairness in wireless networks. The proposed definition of fairness is that, a user has received a fair share of the resources if the user receives a throughput that is equal to or greater than what the user would receive if Generalized Processor Sharing with link quality weights was used as the scheduling discipline. Deng et al. (2009) present a new fairness index, called Fairness Index based on Variational Distance (FIVD) that reflects short-term and long-term fairness at the same time. FIVD calculates the average of the occurrence fairness index that is based on the variational distance of the appearance frequency of different hosts and that of the uniform distribution. Chiu (2000) reviews engineering solutions and economic models for fair allocation of network bandwidth to elastic flows. He explains also the differences between Max/min fairness, Proportional Fairness, and fairness implemented by

Fairness measure	Definition
Max/min	$\max(X_1, \ldots, X_n) / \min(X_1, \ldots, X_n)$
Jain's	$\frac{\left(\sum_{i=1}^{n} X_i\right)^2}{n \sum_{i=1}^{n} X_i^2}$
Relative std. dev.	$\frac{\sqrt{\frac{\sum_{j=1}^{n} \left(X_j - \frac{\sum_{i=1}^{n} X_i}{n}\right)^2}{n}}}{\frac{\sum_{i=1}^{n} X_i}{n}}$
Proportional fairness	If $\sum_{i=1}^{n} \frac{X_i^* - X_i}{X_i} \leq 0$ for any other X^* then X is proportionally fair.

Table 2.1: Definition of some fairness measures

TCP.

Lee et al. (2005) study TCP synchronization and fairness over high-speed networks. They observed that variants of TCP that are specialized for high-speed networks suffer from extreme unfairness, however, in all cases, it was found that use of random drop queuing disciplines eliminates synchronization and the associated unfairness. Zukerman et al. (2005) study trade-offs between efficiency (utilization or revenue) and fairness in a general telecommunications network with relation to any fairness criterion. The authors formulate a linear programming model that finds the optimal bandwidth allocation for a general network under the constraints of α-fairness which is also introduced in that article.

In general, what is considered to be fair depends also on the system under study. Adaptation of an existing fairness measurement method that can be found in the literature to a new specific system is not trivial. The main aim is to find a method for measuring fairness which can be well understood and interpreted. Definition of some important fairness measures are shown in Table 2.1.

In this section, we review two fairness measures in more detail. Jain's fairness index (see Jain et al. (1984)), and the relative standard deviation (RSD). Since we have only positive samples concerning fairness of resource sharing, the relative standard deviation is equal to the coefficient of variation $\left(\frac{\text{standard deviation}}{\text{mean}}\right)$, see Jain et al. (1984). RSD has importance at the problem solved later in the next section (Section 2.2: Fairness-optimal flow initialization). This formula is introduced to show that the fairness-optimal solution is not only valid for a specific fairness measure, but for a wider group of fairness measures. Interestingly, a form of this measure is contained in the formula of the fairness-optimal solution in Section 2.2.

Let $X_1, X_2, \ldots, X_n$ be real positive samples of resource units e.g., flow rate measurements.

The well-known fairness measure defined by Jain et al. (1984) is known as the

Jain's fairness index:

$$J(X_1, \ldots, X_n) \stackrel{\text{def}}{=} \frac{\left(\sum_{i=1}^{n} X_i\right)^2}{n \sum_{i=1}^{n} X_i^2}. \tag{2.1}$$

Jain's fairness index is independent of the population size (e.g., number of users), scale and metric independent, bounded and continuous. This measure has an intuitive meaning; a distribution algorithm with fairness index e.g. 0.1 means that it is unfair to 90% of the users, see Jain et al. (1984). It is easy to show that

$$\frac{1}{n} \leq J(X_1, \ldots, X_n) \leq 1 \ \forall n = 1, 2, \ldots \tag{2.2}$$

The RSD fairness measure (see Jain et al. (1984), [C2]) is defined as follows:

$$\mathrm{RSD}(X_1, \ldots, X_n) \stackrel{\text{def}}{=} \frac{\sqrt{\frac{\sum_{j=1}^{n} \left(X_j - \frac{\sum_{i=1}^{n} X_i}{n}\right)^2}{n}}}{\frac{\sum_{i=1}^{n} X_i}{n}}. \tag{2.3}$$

Note that if $X_1, X_2, \ldots, X_n$ are independent samples from an arbitrary non-negative random variable X, where $EX = \mu$ and $D^2X = \sigma^2$, then (2.3) is the relative standard deviation's estimation of X. It is easy to see that

$$0 \leq \mathrm{RSD}(X_1, \ldots, X_n) \leq \sqrt{n-1}, \ \forall n = 1, 2, \ldots \tag{2.4}$$

There is one-to-one mapping between the RSD fairness measure and Jain's fairness index (Jain et al. (1984)):

$$J(X_1, \ldots, X_n) = \frac{1}{1 + \mathrm{RSD}(X_1, \ldots, X_n)^2}. \tag{2.5}$$

In the case of Jain's fairness index the absolutely fair scenario is assigned 1 value; in the case of the RSD fairness measure it is 0.

Figure 2.2 illustrates the geometrical meaning of the RSD measure in 2 dimensions. Let us consider the case where there are two users (user-1 and user-2) competing for resources. If in a given time instant user-1 gets b_1, and user-2 gets b_2 bandwidth the fair share for both users would be the average of those values ($\bar{b} = \frac{b_1+b_2}{2}$). In Figure 2.2, available bandwidth for for user-2 vs. user-1 is plotted in a given time instant. (b_1, b_2) marks the actual bandwidth allocation of users, $(\bar{b}, \bar{b})$ the ideal and absolute fair allocation. Distance between the actual and ideal points:

$$x = \sqrt{\left(b_1 - \bar{b}\right)^2 + \left(b_2 - \bar{b}\right)^2} = \frac{|b_1 - b_2|}{\sqrt{2}}. \tag{2.6}$$

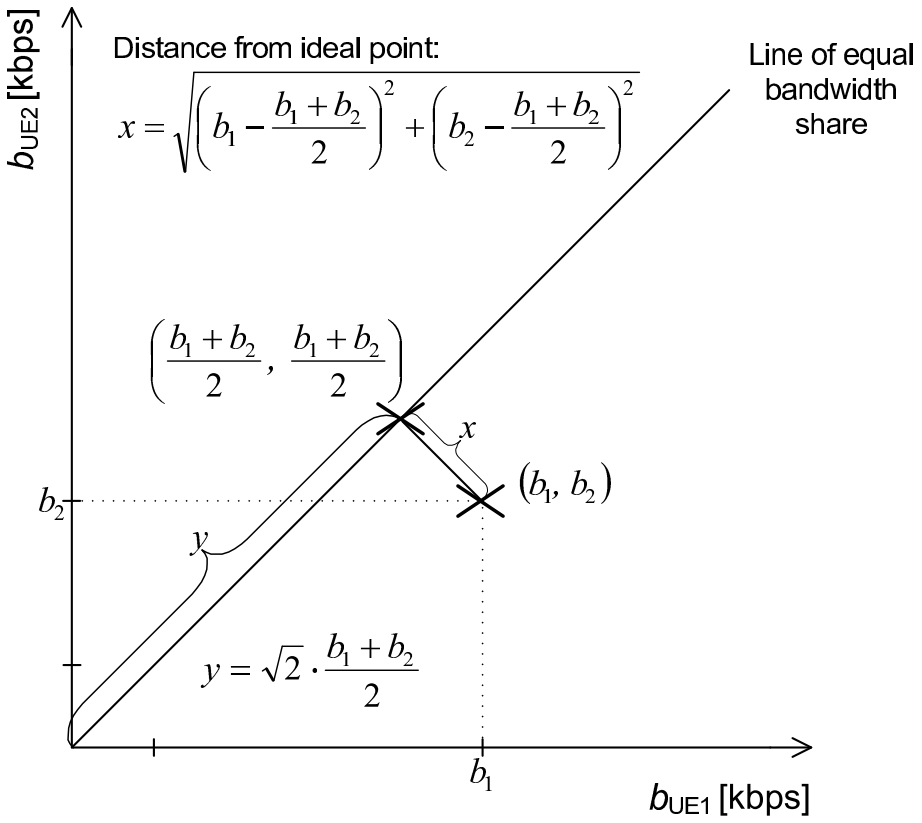

Figure 2.2: The RSD measure in 2 dimensions

Equation (2.6) expresses the difference in kbps from the ideal bandwidth allocation. y denotes the actual distance of the ideal point from the origin of coordinates. It is easy to see that $y = \sqrt{2} \cdot \bar{b}$, and the RSD equals $\frac{x}{y}$ for 2 dimensions.

2.2 Fairness-optimal flow initialization

The proposed method aims to improve fairness independently of the actually used congestion control algorithm by selecting a fairness-optimal initial rate for a new flow. Figure 2.3 illustrates the scope of the method. There are e.g., 2 ongoing flows before 0 sec. Fairness of resource sharing between these two flows is determined by the congestion control algorithm when the time is less than 0 sec. Fairness is changing dynamically in time. At 0 sec, a new flow arrives to the system. If we do not want to modify the congestion control algorithm, this is the only place where we can influence fairness of resource sharing in the system, assuming that the actual rates of ongoing flows are available. We can determine the initial resource share of this new flow at 0 sec. This initial rate should be set to maximize fairness at that time instance. After 0 sec, convergence of fairness of resource sharing is the responsibility of the congestion control again.

In this section we show how to add a new sample (Y) to samples $X_1, X_2, \ldots, X_n$

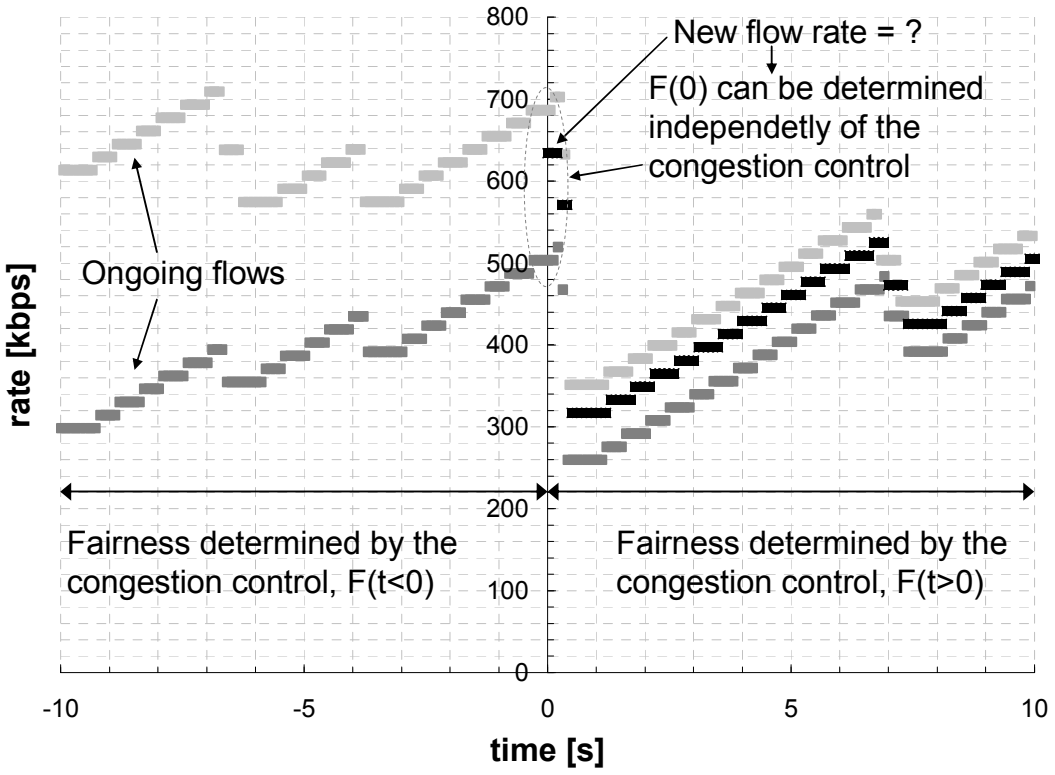

Figure 2.3: Scope of fairness-optimal flow initialization

so that fairness is improved as much as possible. Y represents the initial rate of a new flow, while $X_1, X_2, \ldots, X_n$ are the rates of ongoing flows. Based on the selected fairness measure, the fairness-optimal initial rate can be calculated. E.g., let the selected fairness measure be Jain's fairness index (2.1). The fairness-optimal initial rate of the new flow is the following:

$$Y^* = \arg\max_Y J(X_1, X_2, \ldots, X_n, Y), \tag{2.7}$$

where

$$J(X_1, X_2, \ldots, X_n, Y) = \frac{\left(\sum_{i=1}^n X_i + Y\right)^2}{(n+1)\left(\sum_{i=1}^n X_i^2 + Y^2\right)}. \tag{2.8}$$

$J(X_1, X_2, \ldots, X_n, Y)$ is differentiable with respect to Y and has only one maximum, which occurs at the following point

$$Y^* = \frac{\sum_{i=1}^n X_i^2}{\sum_{i=1}^n X_i} = \frac{\sum_{i=1}^n X_i}{n} + \frac{\frac{\sum_{i=1}^n X_i^2}{n} - \left(\frac{\sum_{i=1}^n X_i}{n}\right)^2}{\frac{\sum_{i=1}^n X_i}{n}}. \tag{2.9}$$

With this value, fairness can be improved to the greatest possible extent. This fairness

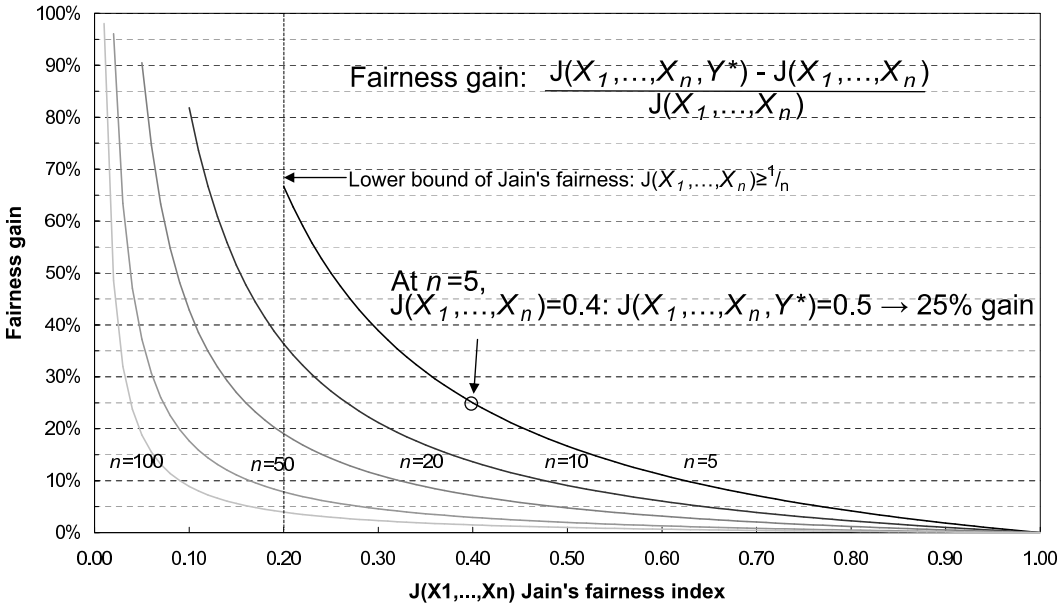

Figure 2.4: Relative increase of fairness (2.10) due to a new fairness-optimal sample

gain depends only on n and $J(X_1, \ldots, X_n)$:

$$J(X_1, \ldots, X_n, Y^*) = J(X_1, \ldots, X_n)\frac{n}{n+1} + \frac{1}{n+1}. \tag{2.10}$$

In Figure 2.4, the relative fairness increase (given by (2.10)) – gained by adding a fairness optimal sample – against current fairness is plotted; e.g. at n=5 if fairness is 0.4 currently it will change to 0.5 after adding Y^*, which means a 25% gain in fairness.

Note that if $X_1, X_2, \ldots, X_n$ are independent samples from an arbitrary non-negative random variable X, where $EX = \mu$ and $D^2X = \sigma^2$, the first additive term in (2.9) is the estimation of X's expected value, and the second is that of its relative variance. The latter is related to formula (2.3), which was our first fairness definition as the relative standard deviation, and here we have the relative variance.

Equation (2.9) can be interpreted as follows. If there is some level of unfairness among the shaping rates of the flows, then the fairness-optimal shaping rate of the new flow is the average shaping rate of the ongoing flows increased with the relative variance of the shaping rates of ongoing flows.

If we used e.g. the RSD fairness measure instead of Jain's fairness index we would get the same result as (2.9), see Section 2.3 for this generalization. Section 2.4 describes how to use these results in an HSDPA system. Section 2.5 presents the validation of the applied results by simulation.

2.3 Fairness-optimal flow initialization for second-order fairness measures

The above described method is not only applicable in the case of Jain's fairness index as presented above, but it is applicable practically independently of the fairness measure (i.e., the method gives the same results) if the measure fulfils certain requirements.

In this section, we show that the fairness optimal solution (2.9) is not only valid in the case of Jain's fairness index, but we get the same result in the case of a larger group of fairness measures. We define a group of second-order fairness measures and prove that the fairness-optimal solution (2.9) holds for them as well. Note that the fairness-optimal solution may be applicable (but may not be calculable) for measures outside this group too, e.g., the Max/min fairness. However, e.g. for the Max/min fairness, it gives a trivial result, since we can choose any values as rate of the new flow between the actual minimum and maximum rates and the fairness of resource sharing would not be influenced. See Figure 2.5 for illustration. In the figure, Max/min fairness index is plotted on a logarithmic scale for the following scenario. 3 ongoing flows (their actual rates: $X_1 = 100\,\text{kbps}$, $X_2 = 200\,\text{kbps}$, $X_3 = 300\,\text{kbps}$) are present in the system at the time instant when the new flow arrives, whose possible initial rates are denoted by Y and plotted on the horizonal axis of Figure 2.5.

2.3.1 Characterization of second-order fairness measures

A fairness measure $F(X_1, \ldots, X_n)$, which can be well used in the engineering practice, should fulfill the following natural requirements defined by Jain et al. (1984):

$$
\begin{aligned}
F(X_1, \ldots, X_n) &= 0 \Leftrightarrow X_1 = X_2 = \cdots = X_n && (2.11)\\
F(X_1, \ldots, X_n) &\geq 0 && (2.12)\\
F(cX_1, \ldots, cX_n) &= F(X_1, \ldots, X_n),\ c \in \Re && (2.13)\\
F(X_1, \ldots, X_n) &= F(X_1, X_1, \ldots, X_n, X_n) && (2.14)\\
F(X_{i_1}, \ldots, X_{i_n}) &= F(X_1, \ldots, X_n) && (2.15)\\
\forall\{i_1, \ldots, i_n\} &\in \prod\{1, \ldots, n\}
\end{aligned}
$$

Requirement (2.11) defines 0 as function value for absolute fairness, i.e., where all samples are the same. Note that in the case of Jain's fairness index this value is 1 instead of 0, but the following results could be derived in a similar way. Equation (2.12) means that the measure should be non-negative. Equation (2.13) ensures that the measure is independent of the applied unit. Scalability in number of samples,

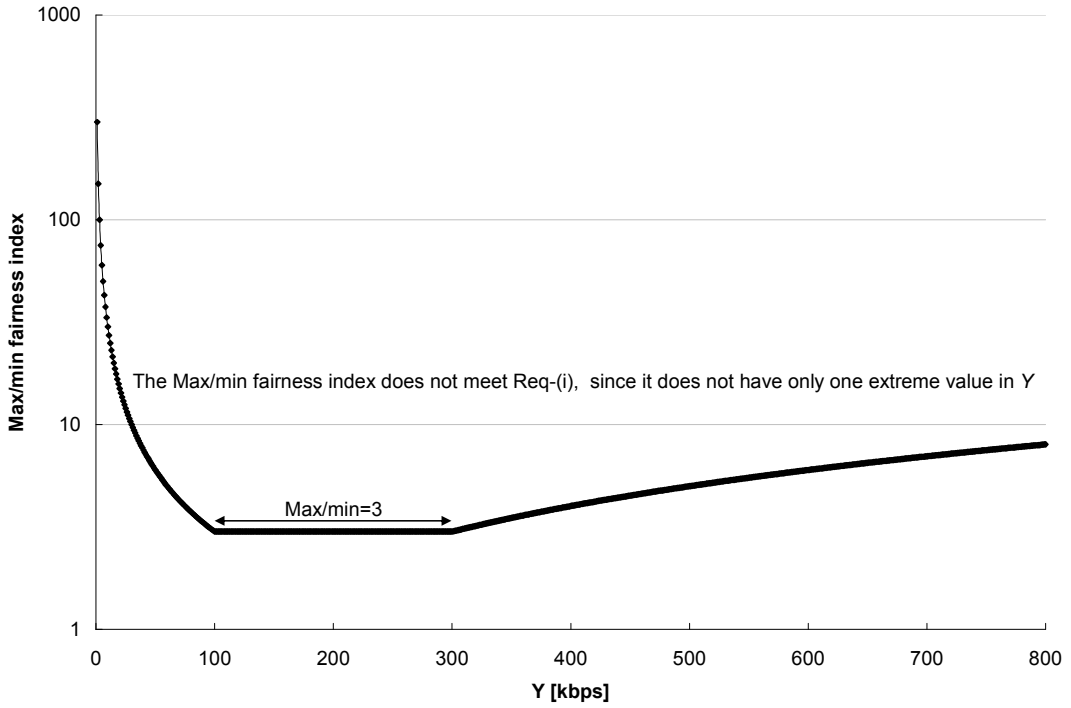

Figure 2.5: Max/min fairness index, $x_1 = 100\,\text{kbps}$, $x_2 = 200\,\text{kbps}$, $x_3 = 300\,\text{kbps}$.

e.g. all samples are doubled, is guaranteed by (2.14). Equation (2.15) means that the fairness function is symmetric, i.e., it does not depend on the order of $X_1, \ldots, X_n$.

For optimizing purposes the following two additional requirements are also useful:

Req-(i): $F(X_1, \ldots, X_n, Y)$ *has only one extreme value in* Y.

Req-(ii): $F(X_1, \ldots, X_n, Y)$ *is differentiable in* Y.

Req-(i) is needed because the fairness optimum should be unambiguous and definite. As a counter-example, we can consider the Max/min fairness where any new Y values between the actual minimum and maximum values of $X_1, \ldots, X_n$ give the same Max/min fairness index (values closer to 1 mean better fairness), see Figure 2.5. It can be seen that the Max/min fairness index does not only have one extreme value in Y, therefore the optimum cannot be determined. Req-(ii) is needed to facilitate optimization. Figure 2.6 illustrates where Max/min fairness index is placed in the set of the fairness functions and how Req-(i) and Req-(ii) modify the considered set of fairness functions.

However, from a mathematical point of view, Req-(i) and Req-(ii) narrow down the set of fairness functions, these requirements can be explained from an engineering point of view, therefore they do not result in loss of generality.

Fairness functions with multiple extreme values cannot be interpreted easily from an engineering and optimization point of view, since they result in more "good" solutions. However, in such cases, we can choose the smallest or the greatest solution in an engineering approach depending on the policy type, i.e. rather a conservative

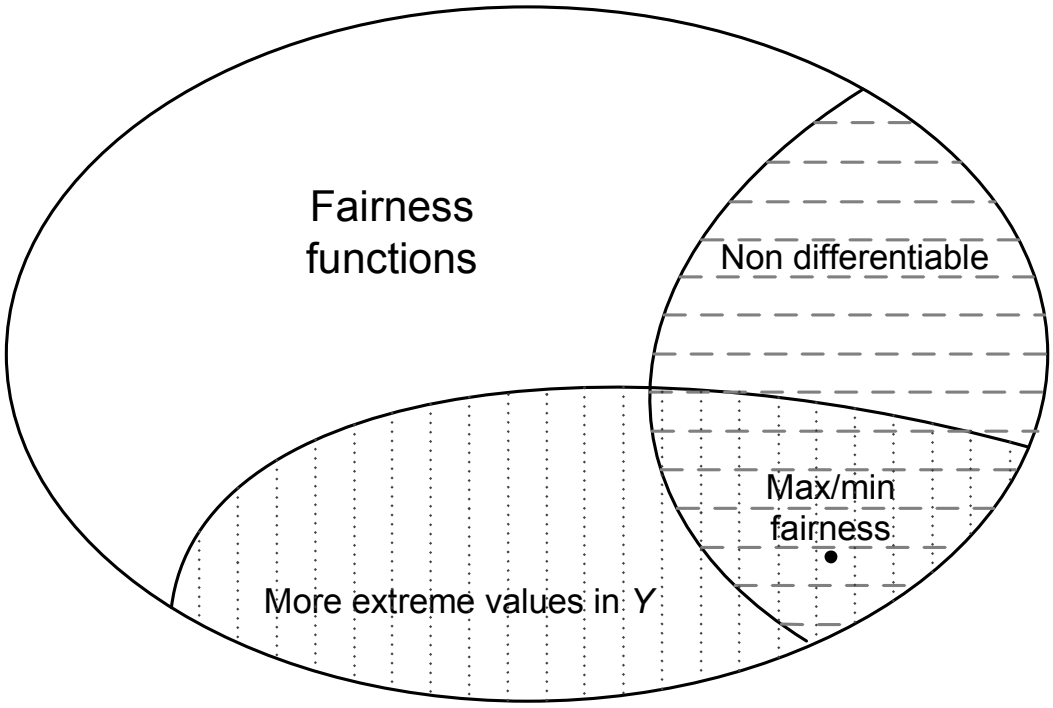

Figure 2.6: Max/min fairness in the set of fairness functions

or a progressive solution. See the Max/min fairness function in Figure 2.5 as an example. According to a conservative policy, Y should be set to 100 kbps, whereas according to a progressive policy it should be 300 kbps, but the value of the fairness index would be the same.

Requiring differentiability narrows down the set of the considered fairness functions to a greater extent both from mathematical and engineering point of view. E.g. it may exclude the absolute value functions or the non-continuous functions. However, non-continuous fairness functions are of smaller importance from an engineering point of view. Continuous functions that are non-differentiable at only some (finite) points (e.g. Max/min fairness) can be handled by finding the optimum individually with analysis. Functions that are non-differentiable on the entire domain are less important from an engineering point of view.

However, the counterexample Max/min fairness is not included in the group of second-order fairness functions to be considered in this chapter. Functions in this group that do not fulfill Req-(i) and Req-(ii) are not exclusively concave or exclusively convex on the entire domain, so they do not fulfil the intuitive requirements of fairness functions; as a consequence they are of small importance from an engineering point of view. Most of the fairness functions – including Jain's fariness index – in the considered group fulfill these two requirements on the considered domain, so these requirements do not result in loss of generality.

We consider such fairness measures that depend on at most second-order statistics,

i.e., $\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}X_iX_j$ and $\sum_{i=1}^{n} d_iX_i$. To fulfill requirement (2.15), for diagonal elements $c_{ii} = c_B$, for non-diagonal elements $c_{ij} + c_{ji} = 2c_A$; and $d_i = d\ \forall i$. Therefore,

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}X_iX_j = c_A\sum_{i=1}^{n}\sum_{j=1}^{n} X_iX_j + (c_B - c_A)\sum_{i=1}^{n} X_i^2 \tag{2.16}$$

and $\sum_{i=1}^{n} d_iX_i = d\sum_{i=1}^{n} X_i$. To fulfill (2.14), $c_A = \frac{c_1}{n^2}$, $c_B = \frac{c_2}{n}$ and $d = \frac{c_3}{n}$, $c_1, c_2, c_3 \in \mathbb{R}$. Note that the first additive term in (2.16) equals $c_A\left(\sum_{i=1}^{n} X_i\right)^2$, consequently, c_A can be zero without loss of generality.

As a consequence, we search $F(X_1, \ldots, X_n)$ in form of

$$g\left(\frac{\sum_{i=1}^{n} X_i}{n}, \frac{\sum_{i=1}^{n} X_i^2}{n}\right).$$

Introducing

$$x = \frac{\sum_{i=1}^{n} X_i}{n},\ \text{and}\ y = \frac{\sum_{i=1}^{n} X_i^2}{n},$$

the requirements (2.11)–(2.15) are as follows:

$$\begin{aligned} x &> 0,\ y \geq x^2 && (2.17)\\ g(x, x^2) &= 0 && (2.18)\\ g(x, y) &> 0 \Leftrightarrow y > x^2 && (2.19)\\ g(x, y) &= g\left(cx, c^2y\right) && (2.20) \end{aligned}$$

Equation (2.17) holds because we have non-negative samples and the variance is also non-negative ($0 \leq \text{Var} = y - x^2$). Equation (2.18) comes from (2.11) and defines absolute fairness. Equation (2.19) comes from (2.12), (2.17) and (2.18). Equation (2.20) comes from (2.13).

From functional equation (2.20) we get the following form of function g:

$$g(x, y) = \Phi\left(\frac{y}{x^2}\right)$$

where $\Phi(\alpha)$ is an arbitrary function (see Aczél & Dhombres (1989)).

Let $\Phi(\alpha)$ be differentiable to complete Req-(ii). Function $\Phi(\alpha)$ should also fulfill the following requirements in accordance with requirements (2.18) and (2.19), respectively:

$$\begin{aligned} \Phi(1) &= 0 && (2.21)\\ \Phi(\alpha) &> 0,\ \text{if}\ \alpha > 1 && (2.22) \end{aligned}$$

The simplest $\Phi(\alpha)$ fulfilling (2.21) and (2.22) is $\Phi_1(\alpha) = \alpha - 1$.

Let $\mathcal{P} = (p_1, p_2, \ldots, p_n)$ be a finite discrete probability distribution and let us define p_i in the following way: $p_i = \frac{X_i}{\sum_{j=1}^{n} X_j}$. The Rényi entropy (defined by Rényi (1960)) of $\mathcal{P}$ can be calculated as follows:

$$\begin{aligned} -\log_2 \sum_{i=1}^{n} p_i^2 &= -\log_2 \sum_{i=1}^{n} \frac{X_i^2}{(\sum_{j=1}^{n} X_j)^2} = -\log_2 \frac{\sum_{i=1}^{n} X_i^2}{(\sum_{i=1}^{n} X_i)^2} = \\ &= -\log_2 \frac{n}{n} \frac{\sum_{i=1}^{n} X_i^2}{(\sum_{i=1}^{n} X_i)^2} = -\log_2 \frac{\alpha}{n}. \end{aligned} \tag{2.23}$$

As a consequence, $\Phi_2(\alpha) = -\log_2 \frac{\alpha}{n}$ is the Rényi entropy of the distribution $\mathcal{P}$ defined above, see also Figure 2.7.

Let us introduce

$$x(Y) = \frac{\sum_{i=1}^{n} X_i + Y}{n+1}, \text{ and } y(Y) = \frac{\sum_{i=1}^{n} X_i^2 + Y^2}{n+1}.$$

Function $\frac{y(Y)}{x^2(Y)}$ *is differentiable and has one extreme value, which occurs at the following point:*

$$Y^* = \frac{\sum_{i=1}^{n} X_i}{n} + \frac{\frac{\sum_{i=1}^{n} X_i^2}{n} - \left(\frac{\sum_{i=1}^{n} X_i}{n}\right)^2}{\frac{\sum_{i=1}^{n} X_i}{n}}. \tag{2.24}$$

Note that (2.24) *equals* (2.9)*. It means that the choice of the fairness-optimal initial rate is independent of the fairness function in the case of the second-order fairness measures introduced above.* For illustration, see Figure 2.7. In this figure, 6 different second-order fairness measures are compared for the same scenario. In the example, 3 ongoing flows (their actual rates: $X_1 = 100\,\text{kbps}$, $X_2 = 200\,\text{kbps}$, $X_3 = 300\,\text{kbps}$) are present in the system at the time instant when the new flow arrives, whose possible initial rates are denoted by Y and plotted on the horizonal axis of Figure 2.7. It is easy to see that the fairness optimum is the same ($Y^* = 233\,\text{kbps}$) for all of the plotted fairness functions.

If $\Phi(\alpha)$ is strictly monotonic on $[1, \infty)$ it keeps the extreme value of $\frac{y(Y)}{x^2(Y)}$, so Req-(i) is fulfilled. If $\Phi(\alpha)$ is not strictly monotonic we have a new local optimum. Consequently, $\Phi(\alpha)$ must be strictly monotonic to fulfill Req-(i).

Such a $\Phi(\alpha)$ for the RSD fairness measure defined by (2.8) is $\Phi_{\text{RSD}}(\alpha) = \sqrt{\alpha - 1}$ and for Jain's fairness index it is $\Phi_J(\alpha) = \frac{1}{\alpha}$. In the latter case, instead of (2.21) $\Phi(1) = 1$ is fulfilled.

In this section we showed that the general form of second-order fairness measures

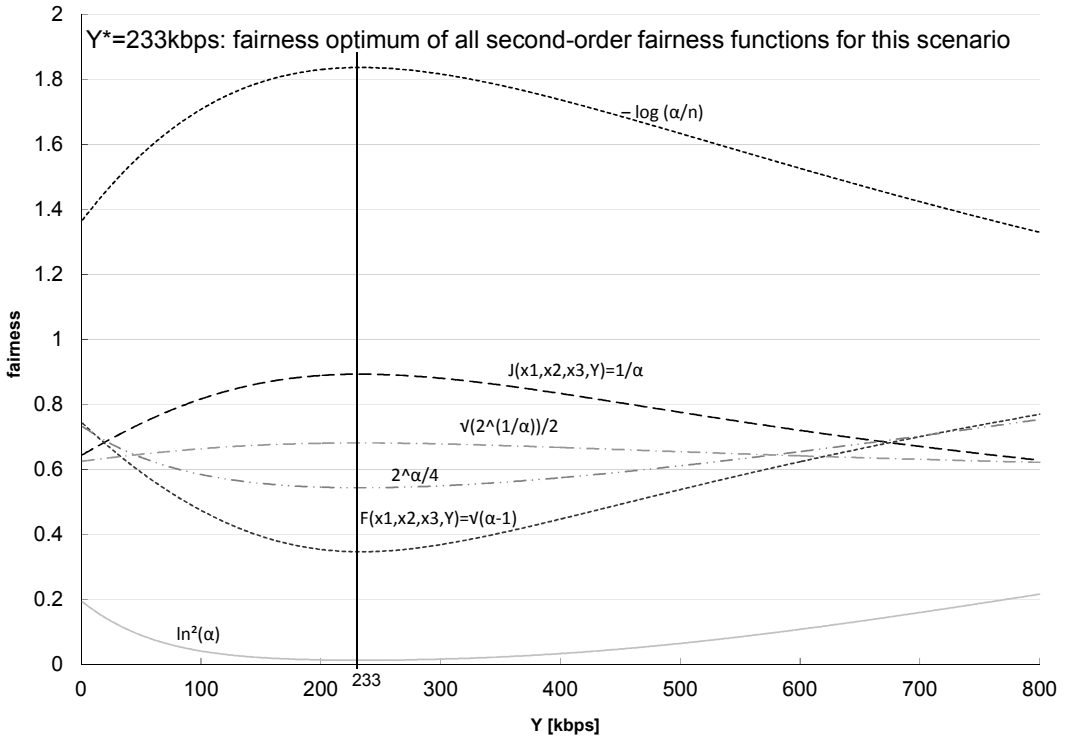

Figure 2.7: Fairness optimum in the case of different second order fairness-measures, $X_1 = 100\,\text{kbps}$, $X_2 = 200\,\text{kbps}$, $X_3 = 300\,\text{kbps}$. $\alpha = 4(X_1^2 + X_2^2 + X_3^2 + Y^2)/(X_1 + X_2 + X_3 + Y)^2$

is

$$F_2(X_1, X_2, \ldots, X_n, n) = \Phi\left(\frac{\frac{\sum_{i=1}^n X_i^2}{n}}{\left(\frac{\sum_{i=1}^n X_i}{n}\right)^2}\right)$$

where $\Phi(\alpha)$ *is strictly monotonic,* $\Phi(1) = 0$, $\Phi(\alpha) > 0$ *if* $\alpha > 1$, *and this measure fulfills requirements of Eq* (2.11)–(2.15). We also gave the general fairness-optimal solution for the introduced second-order fairness measures, which is independent of function Φ.

2.4 Applicability of the results

The proposed method is general enough to be applied in any rate-based congestion control where flows share the same bottleneck. Hereby, we show its application in the RAN Transport Network, more specifically as part of the High-Speed Downlink Packet Access (HSDPA) Transport Network congestion control.

Services using HSDPA do not possess a fixed amount of bandwidth on Transport Network but may use resources not exploited by other services. That is why a congestion control algorithm (see 3GPP (2005a)) is needed on the Transport Network to control the maximum data rate to be used by each flow taking into account the

capacity also which is available on the radio interface (Uu) and the TN.

An important requirement concerning this HSDPA Transport Network Congestion Control algorithm is to share resources among users fairly. Considering fairness among HSDPA flows, two different cases should be distinguished; the case where the Uu interface is limiting HSDPA bandwidth and the case where the TN is limiting. This may change dynamically in time. In the first case, TN capacity is greater than the available Uu capacity. In this case, fairness of resource sharing depends on the radio scheduler algorithms. If the Transport Network is limiting, resources of Uu cannot be maximally utilized. In this case it is the HSDPA TN Congestion Control's task to make sure that TN resources are distributed among users in a fair way, independently of Uu resources. Henceforth, we are focusing on TN-limited scenarios.

The standard by 3GPP (2008b) supports rate- or credit-based congestion control solutions. In this chapter, a rate-based AIMD congestion control is considered as HSDPA Transport Network Congestion Control, as described in Section 1.2.3.2.

2.4.1 The proposed method

The method described in Section 2.2 is realized in the following way. In order to use Fairness-Optimal initial rate (Eq. (2.9)*) from Section 2.2, the rate (X_i) of each flow i currently in the system shall be known. In HSDPA, instead of the actual rates, the actual shaping rates of all ongoing flows are available in the Node B, because they are included in the Capacity Allocation messages to be sent from there to the RNC.*

When a new flow arrives it obviously needs an initial shaping rate. We propose that instead of using a fixed value and/or a Slow Start-like method, the initial shaping rate shall be optimized for fairness of resource sharing, i.e., determined using the latest CA messages of all other ongoing flows which have been sent recently. That is, from shaping rates of all other flows using (2.9) the fairness-optimal initial rate is calculated by the proposed method. If there are no ongoing flows yet, Slow Start shall be applied until the first congestion, afterwards, the AIMD scheme. After determining and sending the fairness-optimal initial rate the AIMD scheme is to be applied.

The proposed method is implemented in Node B, thus we extend the congestion control functionality in the Node B (see Algorithm 1). The contents of the latest Capacity Allocation sent from Node B to the RNC for each flow i is stored. Let c_i denote the shaping rate in the latest CA message for flow i. From these values, the shaping rate c_j can be calculated and used for the new flow j based on (2.9)*. Thus $c_j = Avg(c_i) + \frac{Var(c_i)}{Avg(c_i)}$ $i \in$ set of ongoing flows.*

It is important that all the users executing the algorithm share the same bottleneck because fairness can only be interpreted among them. Thus the method should be performed per bottleneck link if possible. Identification of different bottlenecks is out

of scope of this chapter.

Apart from the Iub interface, the Iur interface which connects two RNCs can also be a potential bottleneck. The Iur interface is used in an inter RNC handover. There, the Serving RNC transmits data via the Iur interface to the Drift RNC, which forwards it to the Node B via its Iub interface. Distinguishing between these bottlenecks is required to select which CA messages should be handled together. If this distinction can be made, the algorithm is to be performed per bottleneck. Otherwise, if the majority shares the same bottleneck, which is a common scenario, and some does not, the algorithm also provides satisfying results.

```
//Variable c[i]: actual shaping rate of flow i
//Existing method: Regular update of ongoing flows' shaping rates
for each flow i
    for every 100ms
    // determine shaping rate c[i] in Bitrate Calculation part
    // based on information from Congestion Detection part
        if congestion detected for flow i
            then decrease c[i]
        else increase c[i]
        Send shaping rates c[i] in CA to RNC
    end for every 100ms
end for each i

//Proposed method (in addition to the existing method):
//At new flow arrival
Input: c[] for all ongoing flows
    if c[] not empty // if there are ongoing flows; calc. the init.
        then c[new flow]=AVG(c[])+VAR(c[])/AVG(c[]) //rate of new flow
    else use slow-start to determine c[new flow]
Output: Shaping rate of the new flow c[new flow]
```

Algorithm 1: Sketch of the method implemented in Node B, see also Figure 1.8

2.5 Evaluation by simulators

In this section, we describe the evaluation of the proposed method by means of a simple AIMD Congestion Control simulator and a complex HSDPA protocol simulator.

2.5.1 AIMD simulator

The proposed method has been evaluated using an AIMD Congestion Control simulator – similar to the one described by Welzl & Mühlhäuser (2003) – to show that

the method can be applied in a rate based congestion control where flows share the same bottleneck. The simulator models flow arrivals and their service. Flows apply bandwidth allocation according to the AIMD scheme. If the sum of actual flow rates exceeds a fixed capacity (C), each flow decreases its rate in a multiplicative way. Otherwise, every flow increases it additively; in this case, linearly.

Flow i has the following parameters. Rate ($r_i[k \cdot t_{\mathrm{RT}}]$) in the k^{th} time slot (RTT, Round Trip Time, t_{RT}), i.e. in $t = k \cdot t_{\mathrm{RT}}$; slot time equals RTT. C denotes the link capacity, i_a the increment of additive increase, and d_m the multiplicative decrease rate.

Flow rates of the next time slot are calculated as follows:

$$r_i[(n+1)t_{\mathrm{RT}}] = \begin{cases} r_i[n \cdot t_{\mathrm{RT}}] + i_a & \text{if } \sum_{\forall j} r_j[n \cdot t_{\mathrm{RT}}] < C, \\ r_i[n \cdot t_{\mathrm{RT}}]/d_m & \text{otherwise.} \end{cases} \tag{2.25}$$

For flows using Slow Start $r_i[(n+1)t_{\mathrm{RT}}]$ is determined in the following way until the first congestion ($\sum_{\forall i} r_i[n \cdot t_{\mathrm{RT}}] \geq C$): $r_i[(n+1)t_{\mathrm{RT}}] = r_i[n \cdot t_{\mathrm{RT}}] \cdot s_r$, where s_r denotes the Slow Start increase rate. Initial rate of a new flow f is determined for fairness-optimal users in the following way according to (2.9):
$r_f^*[(n+1)t_{\mathrm{RT}}] = \frac{\sum_{\forall i} r_i[n \cdot t_{\mathrm{RT}}]^2}{\sum_{\forall i} r_i[n \cdot t_{\mathrm{RT}}]}$.

AIMD does not define the start of a flow. The scheme deals only with the dynamic behaviour of flows, after congestion. The AIMD Congestion Control under study has two extensions. It either executes Slow Start in the beginning, or applies the fairness-optimal method described in Section 2.2.

Simulation results also proved that AIMD Congestion Control with fairness-optimal initial rates (Section 2.2) provides – as expected – better fairness than with fixed initial rates whereas link utilization does not decrease. The proposed method has also been compared to an extended AIMD scheme using Slow Start until the first congestion for reaching the fair bandwidth share. The proposed method outperforms this scheme also in terms of fairness while maintaining the same utilization level.

In Figure 2.8, the transient behavior of the Fairness-Optimal method versus Slow Start is visualized. 16 kbps is used as Slow Start's initial rate, $C = 1$ Mbps, $i_a = 16$ kbps, $d_m = 2$, $s_r = 2$, $t_{\mathrm{RT}} = 1$. The upper two charts in this figure show the changing of bandwidth per user vs. time, whereas the one at the bottom visualizes fairness in time. The upper figure illustrates four Slow Start-like users and the middle figure two Slow Start-like plus two Fairness-Optimal ones. Circles mark Slow Start-like user arrival, and dots mark Fairness-Optimal ones. Until the first Fairness-Optimal user arrives (at 1.6 sec) both figures are the same. At this time instant the system is not in equilibrium yet. In the upper figure a Slow Start-like

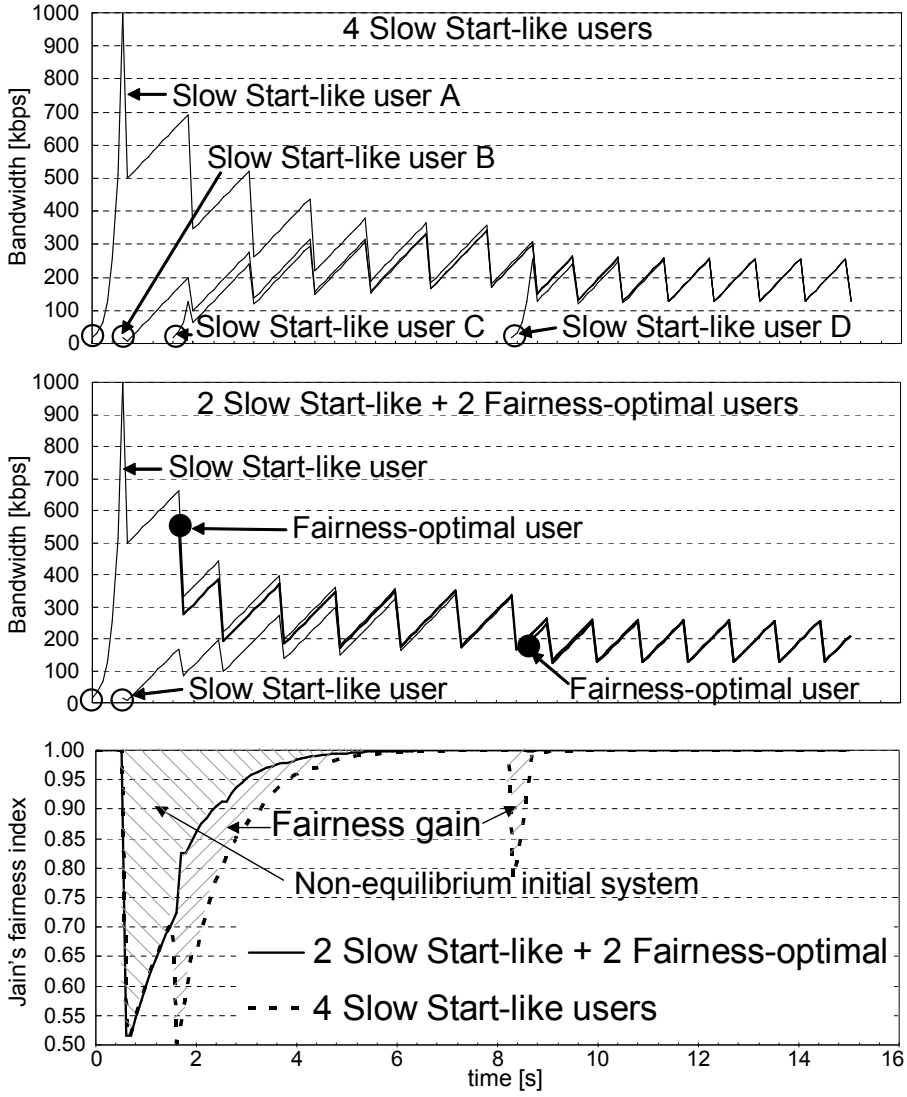

Figure 2.8: Transient fairness at Slow Start and Fairness-Optimal

user arrives and in the middle figure a Fairness-Optimal one. As it can be seen the Fairness-Optimal user arrival improves fairness significantly, whereas the Slow Start-like method worsens it drastically (approx. 20%, e.g. at 1.8 sec). The gain in fairness can be seen in the bottom of the figure (striped area). However, fairness in both cases converges. At 8.2 sec, after the system has reached its equilibrium (fairness index = 1), user arrival occurs again. In the upper figure Slow Start-like and in the middle figure Fairness-Optimal. The former worsens fairness to approx. 20% again, while the latter does not degrade it at all.

The benefits of the algorithm can be obviously seen, however the algorithm in itself cannot be further analyzed in this environment because modelling of the natural unfairness, which is present in real systems, is not feasible. In real systems, flows do not always experience congestion at the same time. If all incoming flows were started

using the Fairness-Optimal method, an ideal system would be reached. The system would work synchronously granting each flow equal bandwidth, including new flow arrivals also. In order to study the algorithm in more detail, a more sophisticated simulator is needed.

As a conclusion, Slow Start works properly, but it is not optimal. The Fairness-Optimal method performs well based on this model. The present simulator is not capable of more detailed analysis because of modelling limitations, therefore the mentioned further analysis is needed.

2.5.2 HSDPA protocol simulator

The proposed method has also been implemented in a 3G/HSPA protocol simulator. The simulator contains HSDPA related protocol functions, such as TCP/IP, Radio Link Control (RLC), Medium Access Control – dedicated (MAC-d), High-Speed Downlink Shared Channel (HS-DSCH) Iub framing and Medium Access Control – high-speed (MAC-hs). The ATM/AAL2 (Asynchronous Transfer Mode/ATM Adaptation Layer 2) Transport Network is modelled as a link of fixed capacity with a finite buffer, and fixed downlink propagation delay. The radio environment consists of standard models for distance attenuation, shadow fading and multipath fading, based on 3GPP typical urban channel model, see 3GPP (2005b). The radio scheduler in Node B uses round robin scheduling scheme. The radio network used by the simulator consists of an RNC and a Node B with three cells.

2.5.2.1 Long-term fairness and aggregated throughput comparison of the proposed method and Slow Start

In order to illustrate the benefits of the proposed method, the following scenario has been built up. As it is visualized in Figure 2.9 and 2.10, at the presence of two long background flows (gray lines) short flows join and leave the system (black lines) frequently.

In both figures, *aggregated* TCP throughput is plotted. It means that the dark gray line represents the sum of the first long background flow's TCP throughput (light gray line) and the second long background flow's TCP throughput. Likewise, the black line represents the sum of both long background flows' TCP throughput (dark gray line) and the short flows' TCP throughput.

Transport Network capacity is 2 Mbps, buffer size is 50 ms. Slow Start initial rate is 32 kbps.

In Figure 2.11 Jain's fairness index is plotted vs. time. We can see that at all user arrivals, fairness decreases significantly because of the Slow Start mechanism. Slow Start (Figure 2.9) may not be going to reach the fair bandwidth share in the short run

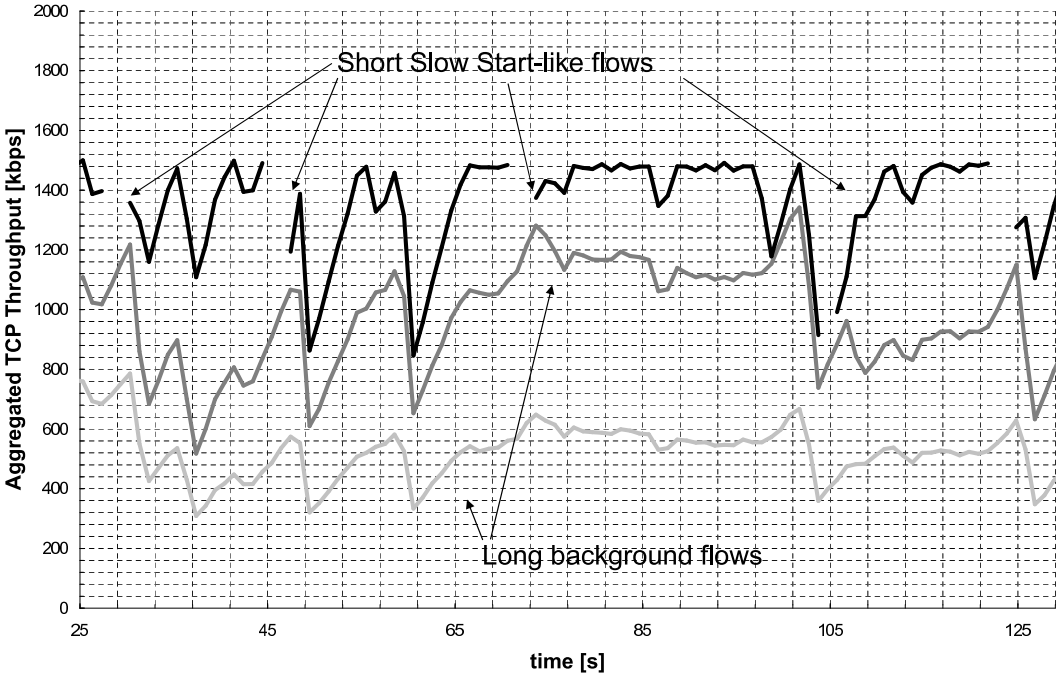

Figure 2.9: Throughput without the Fairness-Optimal method

because of other flows, which provoke congestion before the new flow could reach its fair share. That is, the new flow at the beginning of the Slow Start phase should also halve its rate (e.g. at 30 sec in Figure 2.9). This is why the new flow remains at the starting point of Slow Start for a longer time period, whereas other flows have higher bitrate. This results in unfairness at user arrivals. In Figure 2.11, it can be noticed that average fairness in the case of the Fairness-Optimal method is also significantly improved. Slight worsening in transient fairness in the case of the Fairness-Optimal method may occur due to interactions of other protocols (e.g. TCP).

2.5.2.2 Short-term fairness and shaping rate comparison of the proposed method and Slow Start

Figure 2.12 focuses on a particular user arrival from a scenario similar to the ones shown in Figure 2.9 and 2.10. A 200-ms TN buffer is used and instead of the aggregated TCP throughput, the shaping rate is shown, which is included in the CA message sent from Node B to RNC (see Figure 1.8).

Figure 2.13 shows the same scenario but at user arrival the fairness-optimal method is used. Both in Figure 2.12 and 2.13, the shaping rate is plotted on the primary axis and Jain's fairness index on the secondary axis against time.

It is noticeable that the period of temporary unfairness caused by the new user arrival is much shorter in the case of fairness-optimal initial rate. The fairness-optimal method gives the new user a bandwidth which does not worsen fairness. It is close to the actual rates that the present users have (2.9). In most cases (as it can be seen in

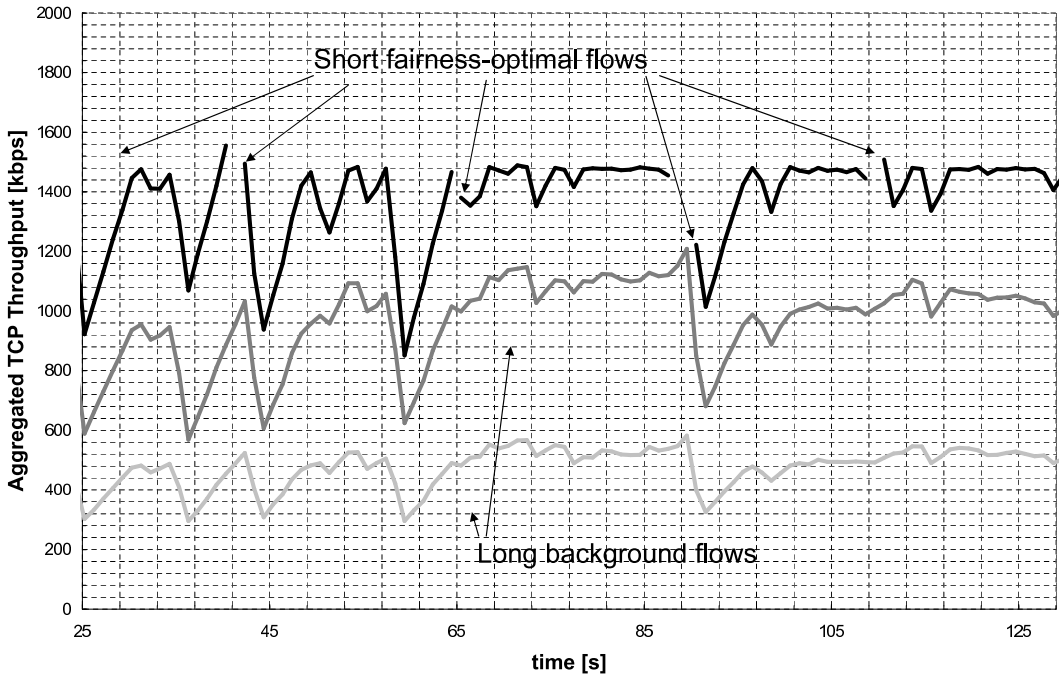

Figure 2.10: Throughput of Fairness-Optimal method

Figure 2.13 too), it causes congestion immediately so the stationary phase of AIMD is reached with a 'fair' bandwidth allocation of the flows. Slight worsening in transient fairness may occur due to interactions of other protocols.

Beside the TCP-level fairness that was plotted in Figure 2.12 also, fairness of the shaping rates (CA rates) are also visualized. It can be noticed that the arrival of the fairness-optimal flow even improves fairness of the system on CA level (like in Section 2.5.1). Negligible worsening of fairness (10% for 100 ms) can be noticed after the first congestion (not upon user arrival) because flows do not react in exactly the same time. In contrast, in Figure 2.12 the Slow Start phase causes unfairness in both CA and TCP level. The time period of temporary unfairness lasts much longer compared to the previous case. This increases unfairness on average.

2.5.2.3 Average-fairness comparison of the proposed method and other methods

In order to show that the proposed method not only outperforms the Slow Start-like one in individual cases, several scenarios have been investigated. In Figure 2.14, TCP-level average fairness against the number of users is plotted. Average fairness includes both transient fairness degradation and convergence speed. The TN capacity is 4 Mbps, 100-kB files are being downloaded periodically, the TN buffer is 200 ms. Without using the proposed algorithm ($b.$,), fairness decreases with the number of users. In contrast, the proposed method ($d.$,) assures excellent fairness (more than 0.9) independently of the number of users. Two additional methods were also investigated

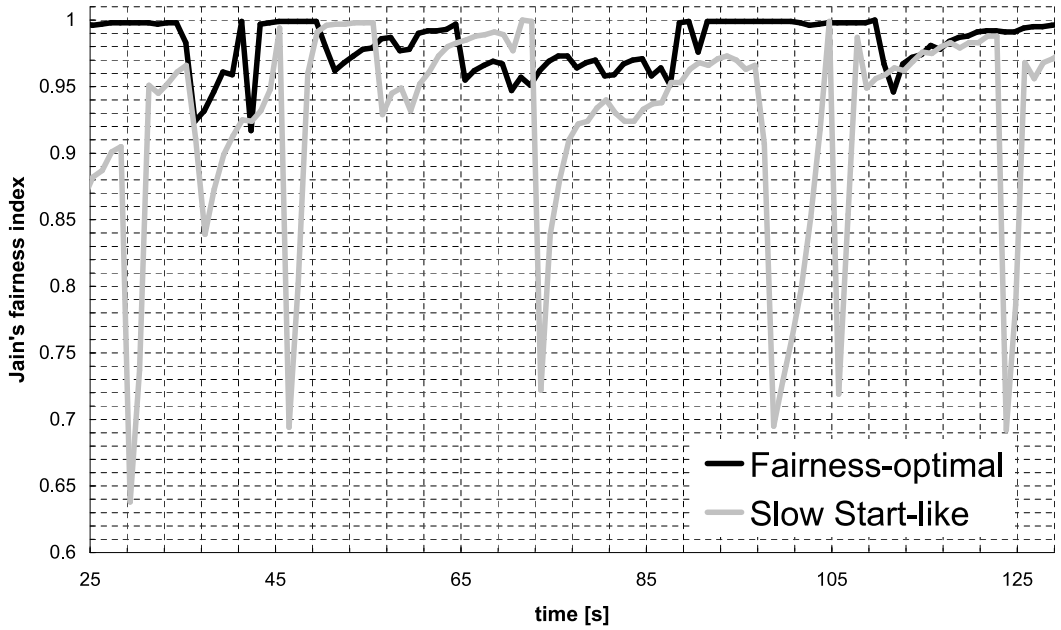

Figure 2.11: TCP-level fairness with and without the Fairness-Optimal method

as references. At the first one, flows' shaping rate start from the TN capacity ($a.$,) at the second one from the current average CA rate ($c.$,). In the chart we can see that fairness (light & dark gray columns) also decreases with the number of users using both methods. However, they both perform better in the beginning than the Slow Start-like method, but at large number of users (18–20) the Slow Start-like method outperforms method a. Method c proves to be the second best solution, which can also be seen in Table 2.2, where minimum, average and maximum fairness gains of the Fairness-Optimal method (d) compared to methods a, b, and c are shown. This table also shows that the maximal fairness gain occurs in the case of method a (method a scales the worst in number of users), and the largest average gain occurs in the case of method b.

As a summary, the Fairness-Optimal method provides fairness uniformly, independently of the number of users and assures faster convergence to fairness than starting from average. We have also shown that the long-term average fairness is improved up to 30%, on average 20% compared to slow start, see Table 2.2. The recommended method not only improves fairness but it does not decrease throughput either.

The benefits of the proposed method are also observable assuming periodical downloads of large files but not as significantly as in the case of web objects. Because of the long download periods, AIMD has enough time to reach fairness and the fairness-optimal initial rate is less frequently used. The smaller the files are the more often the algorithm is applied so the bigger impact it has on average fairness.

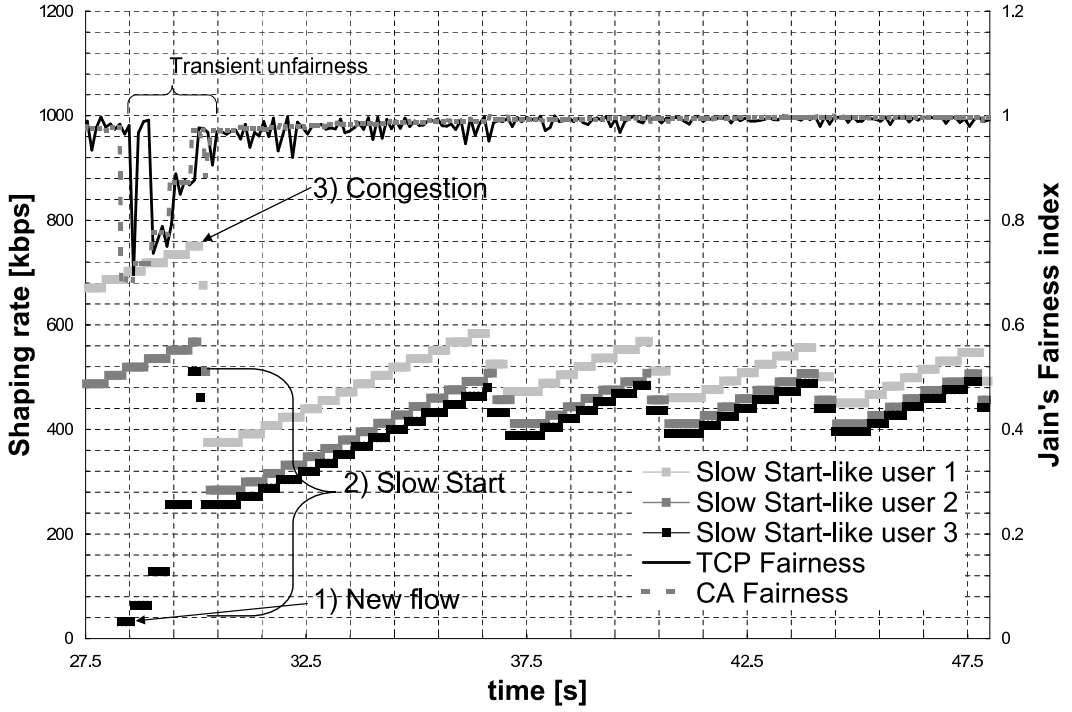

Figure 2.12: Transient fairness without the Fairness-optimal method

The proposed method (d) compared to	minimum fairness gain	average fairness gain	maximum fairness gain
a.) Starting from link capacity	0%	11%	38%
b.) Slow Start-like method	2%	20%	31%
c.) Starting from average rate	0%	6%	15%

Table 2.2: Fairness gain of the Fairness-Optimal method compared to other methods

2.5.3 Conclusion

The proposed method has been evaluated by means of two different kinds of simulators. The first one, the AIMD Congestion Control simulator – modelling an ideal system – shows that the proposed method has significant benefits in terms of improving fairness. These benefits are also shown by the HSDPA protocol simulator. However, in this case they are less pronounced due to asynchronous congestion signals, interactions of more protocols and environment parameters.

2.6 Conclusion

A method which provides fairness-optimal initial shaping rate for incoming HSDPA flows based on ongoing HSDPA flow rates has been proposed. It has been shown by

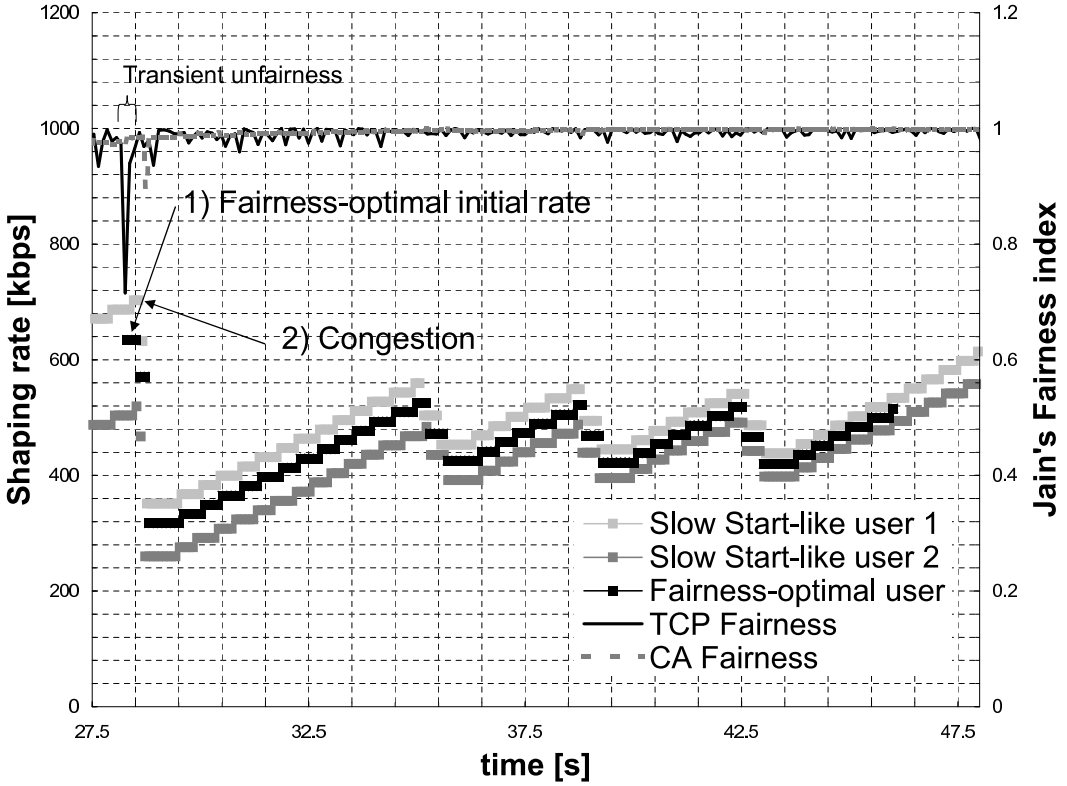

Figure 2.13: Transient fairness with the Fairness-optimal method

simulations that the proposed method improves the transient fairness characteristics at user arrivals and affects the long-term average fairness in an advantageous way also, providing good fairness uniformly. The method can be applied in a rate-based or window-based congestion control where flows share the same bottleneck. A general solution for fairness-optimal initial rate was also given in the case of second-order fairness measures.

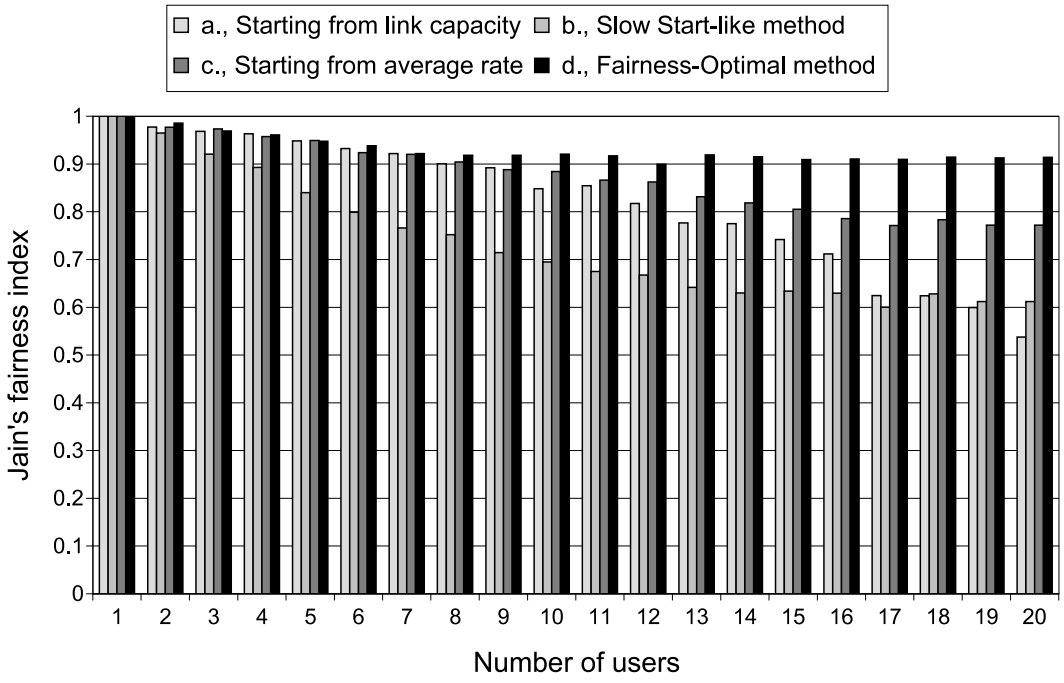

Figure 2.14: TCP-level average fairness

Chapter 3

Window-based RAN Transport Network Congestion Control

Since the RAN Transport Network may be a potential bottleneck, congestion control is needed. In the previous chapter an extension for a rate-based congestion control was presented. This chapter presents a novel window-based HSDPA Transport Network congestion control. This solution is original because in the transport network mainly rate-based congestion control methods are used. The novelty in this method is the extension of the RLC protocol with congestion control functionality.

Considering the protocols in the RAN Transport Network, an interesting duality can be noticed between TCP and RLC. TCP was originally designed to handle loss as an indicator of congestion, i.e., if loss occurs rate reduction is needed. However, loss due to radio link failures (that are typically not because of congestion; but because of, e.g., handover or temporarily poor radio conditions) should be handled in a different way, otherwise TCP is not efficient over wireless networks.

On contrary, the original purpose of the RLC protocol is just the very opposite. RLC was designed to handle loss and increased delay due to radio link failures only, i.e., data units that are suspected to be lost are simply resent. Loss or delay increase due to congestion should be handled in a different way, otherwise RLC does not work efficiently over congested wired links such as the transport network in the case HSDPA.

As network conditions have changed, wireless networks evolved, many extensions have been proposed to improve TCP performance over radio links, see Assaad et al. (2004). Similarly, 3G/HSDPA network conditions have also changed (i.e., Iub became a bottleneck and therefore congestion may occur in the RLC loop), thus the new shortcomings of the RLC protocol should be handled somehow. Existing solutions, such as resolving congestion below RLC in a separated layer (see, e.g., Nádas et al. (2007)) are efficient solutions, but a different approach is investigated here. In this

chapter, we study how the RLC can be extended to be used over congested wired links.

The main idea behind the proposed extension is to control the RLC transmission window size. This transmission window is originally part of the protocol but its size is fixed during a connection. A sophisticated method is proposed in which the size of the transmission window is adjusted according to the level of congestion, in order to control congestion.

Research results related to this chapter are published in [C1] and [J3]. My theses are related to all the main results of this chapter.

3.1 Introduction

In this section, an overview is given about the challenges of the Radio Access Network, the environment where the proposed method is implemented. In this network, there may be more potential bottlenecks, and each of these needs different type of handling methods. The most important Transport Network specific protocols, such as the Iub Framing Protocol and the Radio Link Control (RLC) protocol is also reviewed. The acknowledged mode RLC protocol makes the congestion control more challenging in this environment because of its retransmission mechanisms. This issue is also addressed in the next subsections.

3.1.1 Bottlenecks in the HSDPA system

In the HSDPA system, there may be two potential bottlenecks; the air interface (Uu) and the RAN Transport Network. Different handling is needed for both of them.

The air interface bottleneck is basically handled by the Hybrid Automatic Repeat reQuest (HARQ) protocol along with fast radio link adaptation, soft-combining of partially received data blocks and fast channel-dependent scheduling, see Dahlman et al. (2008) for further details. These methods result in a very small residual error probability on the air interface. The residual error is corrected by the retransmission mechanisms of the Acknowledged Mode RLC protocol.

The Transport Network may already become a potential bottleneck with the introduction of HSDPA, since for HSDPA flows no resource reservation is made. For Dedicated Channels (DCHs), transport network bandwidth can be reserved by means of admission control, whereas bandwidth reservation is not efficient for HSDPA, because of the higher data rates and much higher variance of achieved bitrate. In the Internet such congestion situations are handled by e.g. the end-user TCP. However, this is not possible in the transport network, because lost packets are always retransmitted by the Acknowledged Mode RLC protocol, so TCP cannot use packet loss

as indication of congestion. As a consequence, a new solution is needed to control congestion.

3.1.2 The Radio Link Control Protocol

The acknowledged retransmission mechanism of the Radio Link Control (RLC) protocol ensures reliable transmission of loss-sensitive traffic over the air interface. The normal RLC mode for packet-type services is the acknowledged mode, so RLC works in this mode in the case of HSDPA, too. RLC Acknowledged Mode always retransmits lost PDUs and it does not include congestion control functionality. As a consequence it cannot handle congestion situations and it does not let TCP handle congestion either.

The RLC status-report mechanism is controlled by pre-configured triggers (i.e., timers and/or counters) determining the status sending period. In practice, status messages containing ACKs are sent periodically, relatively seldom. If there are NACKs in the status message, the sender retransmits the missing PDUs giving them priority over new ones. Several unsuccessful retransmissions of the same PDU trigger an RLC reset and the whole RLC transmission window is discarded. Meanwhile, TCP is not informed about the congestion situation. As a consequence of the periodical status sending, traffic may become bursty without shaping.

3.1.2.1 RLC summary

In this section, we review features of the RLC protocol that were found to be important during the analysis, further details can be found in the standard by 3GPP (2009).

RLC provides three types of service to the upper layer according to the standard: transparent (no additional protocol information), unacknowledged (delivery not guaranteed), and acknowledged data transfer mode. The normal RLC mode for packet-type services is the acknowledged mode, so RLC works in this mode in the case HSDPA as well. The acknowledged retransmission mechanism of the radio link control (RLC) protocol ensures reliable transmission of loss-sensitive traffic over the air interface. The RLC protocol is used by both signaling radio bearers and radio bearers for packet-switched data services, but not by radio bearers for circuit-switched services.

RLC Acknowledged Mode (AM), which is a Selective Repeat Automatic Repeat-request (SR-ARQ) protocol, provides transport service to upper layers between UE and the RNC. RLC AM does not include congestion control functionality.

The RLC status reports containing ACKs/NACKs, which are being sent regularly based on preset events, trigger retransmission of all missing PDUs. The time interval

between two status sending events is predefined at the receiver side, which dominantly determines the periodical status sending period. The receiver side detects missing RLC PDUs based on gaps in the sequence numbers. The receiver side RLC requests retransmission by sending back a status PDU, informing which PDUs within the receiving window have been acknowledged (ACK) or lost (negative ACK, NACK). Upon reception of a status message, the sender can slide its transmission window (Tx window) if one or more in-sequence frames are acknowledged, so that new PDUs can be sent. If there are NACKs in the status message, the sender retransmits the missing PDUs giving them priority over new ones. Several unsuccessful retransmissions of the same PDU trigger an RLC reset and the whole RLC Tx window is discarded, see Section 3.1.4.1.

The Polling function is used by the Sender to request the peer RLC entity for a status report. The "Polling bit" in the Acknowledged Mode Data (AMD) PDU or the POLL super-field in the downlink indicate the poll request. There are several triggers for initiating the Polling function. Which of the triggers shall be used is configured by upper layers for each RLC entity. The following triggers can be configured:

- Last PDU in buffer.
- Last PDU in retransmission buffer.
- Poll timer. When the Poll Timer expires the Sender triggers the Polling function.
- After a certain number of sent PDUs.
- After a certain number of sent SDUs.
- Window based. A poll is triggered for each AMD PDU when the Poll Window is not greater than a certain percentage of the transmission window.

Each status report consists of one or several STATUS PDUs. The Receiver shall trigger the transmission of a status report when receiving a poll request. Additionally, the following triggers for transmission of status reports are configurable by upper layers, according to the standard:

- Detection of missing PDU(s). If the Receiver detects one or several missing AMD PDUs it shall trigger the transmission of a status report to the Sender.
- Timer based status report transfer. The Receiver triggers the transmission of a status report to the Sender periodically.

The Receiver is not allowed to transmit a status report while acknowledgement is prohibited even if any of the triggering conditions above are fulfilled. If a status

report was triggered during this time, the status report is transmitted after the timer Timer_Status_Prohibit has expired.

AMD PDUs buffered in the Retransmission buffer are deleted or retransmitted based on the status report found within a STATUS PDU or Piggybacked STATUS PDU sent by the peer AM RLC entity. This status report may contain positive or negative acknowledgements of individual AMD PDUs received by the peer AM RLC entity.

The most important timers that we use in the implementation of the proposed method are the following ones:

- Timer_Status_Prohibit. It is meant to prohibit the Receiver from sending consecutive acknowledgement status reports. In the UE, this timer shall be started (or restarted) when the transmission of the last STATUS PDU of an acknowledgement status report is indicated by lower layer. From the time an acknowledgement status report is triggered until the Timer_Status_Prohibit timer expires, acknowledgement is prohibited.

- Timer_Status_Periodic. This timer enables timer based status reporting. This timer shall be started when the RLC entity is created. When the timer expires the transmission of a status report shall be triggered and the timer shall be restarted. This timer can be blocked by upper layers. The timer shall be restarted when upper layers indicate that it is no longer blocked.

3.1.3 Congestion Detection

The 3GPP-standard congestion detection is implemented in the Iub Framing Protocol layer in the Node B, see Figure 3.1. Each flow has its own congestion control and congestion detection functionality.

Whenever an HS-DSCH Data Frame arrives from the RNC to the Node B congestion detection is performed. Congestion detection is based on the Frame Sequence Number (FSN) and Delay Reference Time (DRT) fields of the HS-DSCH Data Frame, see 3GPP (2008c). FSN is used to detect missing data and DRT is used for dynamic delay measurements. Data loss and increased delay on the Transport Network may be the indicators of TN congestion. The HS-DSCH Capacity Allocation (CA) Control Frame, which is sent from the Node B to the RNC, is used to inform the RNC about the Transport Network conditions. The CA message contains the 2-bit Congestion Status flag to indicate the severity of Transport Network congestion. If the detected level of the congestion is high, the Congestion Status field in a Capacity Allocation Control Frame will be filled with the *Loss* flag. If the result of the detection is a lower

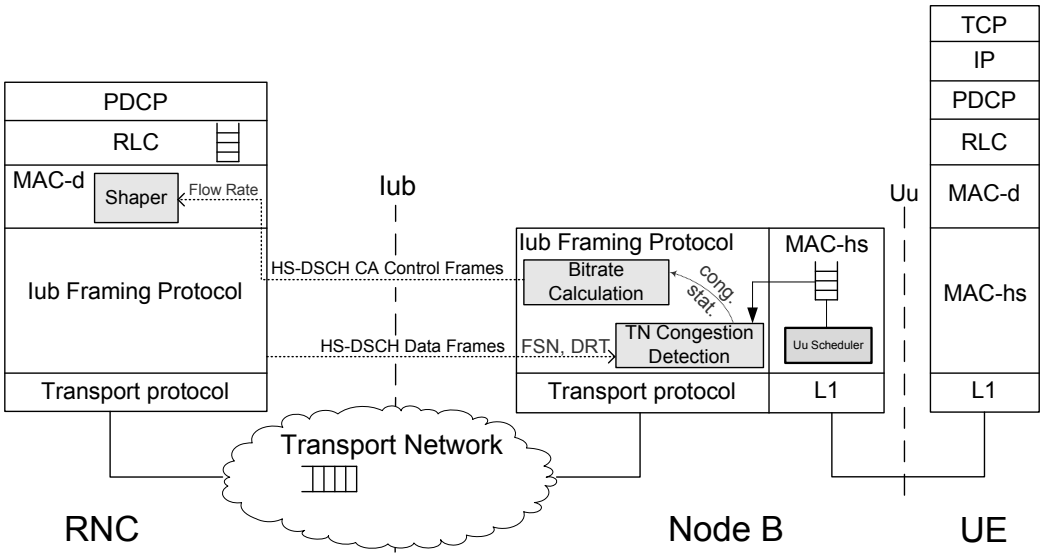

Figure 3.1: HSDPA specific protocol stack with the 3GPP congestion control and congestion detection architecture

congestion level, the Congestion Status is filled with a *Delay* flag. After that, the CA will be transmitted to the RNC.

For a fair congestion control, a 'fair' congestion detection method is crucial. However, even if all the flows share the same bottleneck queue they do not experience and detect exactly the same delays or losses, so we cannot always assume synchronous congestion signals. This is because of sampling issues, since congestion detection only has samples when it has received data frames in Node B. That is why bursty (i.e., packets belonging to different flows are not mixed enough with each other) traffic spoils fairness of congestion detection. Therefore, congestion detection is challenging in this environment.

3.1.4 Problems with the lack of congestion control

Without congestion control, high Transport Network loss and serious congestion occurs.

TCP slow start normally increases the TCP congestion window (cwnd) size fast to its maximum and it is kept at that value, because due to RLC retransmissions, TCP does not experience loss (except for e.g. RLC reset). In the case of Transport Network congestion, RLC keeps retransmitting lost PDUs until they are successfully received (or until RLC resets), however these retransmissions even increase the congestion level. In this way, TCP is significantly later notified about the Transport Network congestion. As a summary, we need a system specific congestion control (CC), because the RLC does not have congestion control functionality and TCP congestion control can not operate efficiently above the RLC AM protocol.

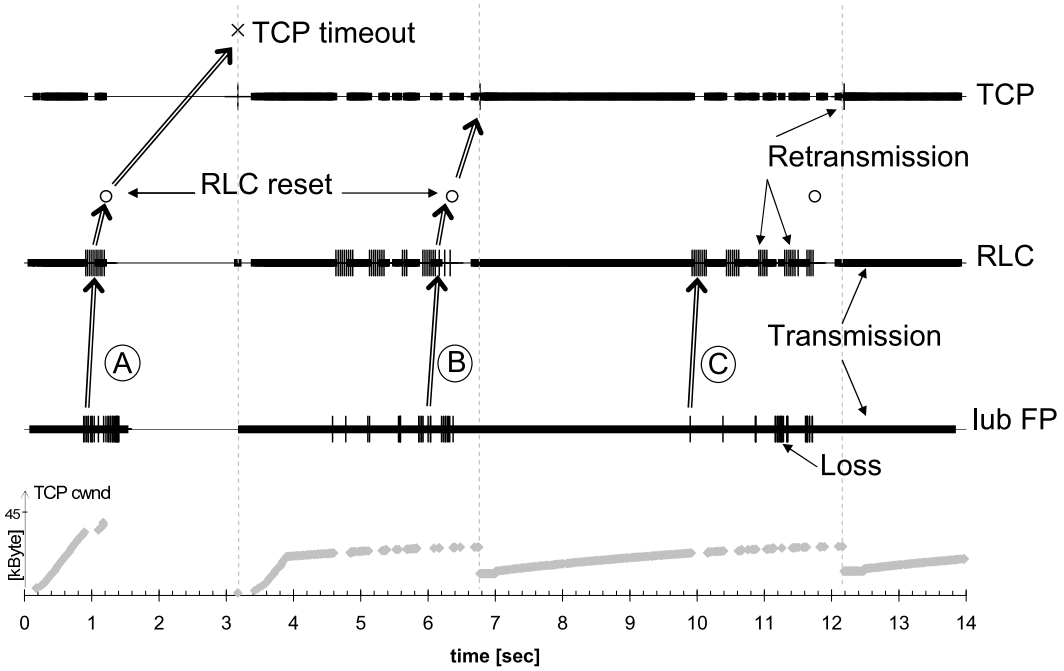

Figure 3.2: TCP and RLC behavior without TN congestion control

3.1.4.1 HSDPA without congestion control

In this section, it is shown why Transport Network congestion control is needed for HSDPA. Consequences of the lack of TN congestion control in terms of TCP and RLC protocol behavior are illustrated in Figure 3.2. In this example, Iub TN bottleneck capacity is 1 Mbps, TN buffer size is 200 kbit, one user downloading a 1 MB file is present in the system and no TN congestion control is applied. Note that in the case of more users, protocol behavior without congestion control is even worse. The RLC PDU size is 336 bits, RLC maximal window size is 2047 PDUs, RLC status sending interval is 10 ms, RLC reset threshold is 30 retransmissions and TCP maximal cwnd is 262 kBytes. On the top of the figure, TCP and RLC transmissions (▪), retransmissions (|), TCP timeouts (x) and RLC resets (o) are plotted against time. Vertical lines (|) on the Iub FP line represent data frame (DF) loss due to TN buffer overflow. At the bottom, TCP congestion window size vs. time is plotted. The lack of TN congestion control means that the RNC wraps arriving MAC-d PDUs into DFs immediately and forwards them to the Node B by putting them into the TN.

In Figure 3.2 three cases (A, B and C) are presented. A relatively advantageous scenario is depicted in this example because the worst case (A) occurs only in the beginning.

In case A, congestion causes TCP Retransmission Timeout (RTO), i.e. the RTO timer expires. The fast bitrate increment during TCP Slow Start overloads the TN causing TN buffer overflow at 1 sec. The consequent Data Frame loss provokes several RLC PDU retransmissions that even increase the overload. RLC only retransmits

but does not have TCP-like congestion handling. A large number of unsuccessful retransmissions triggers an RLC reset and the whole RLC Tx window containing all unacknowledged PDUs is discarded, whereas all impacted SDUs are dropped. Meanwhile, the congestion situation is hidden from TCP, i.e., no new TCP ACKs arrive because all RLC SDUs were dropped. The lack of ACKs causes RTO to expire at 3.2 sec and cwnd size is set to its minimum.

In case B, overload causes RLC reset only. At 6 sec, loss on FP level due to TCP's linear bitrate increase in congestion avoidance also results in RLC retransmissions and reset. However, only a TCP retransmission is triggered because a luckily arrived TCP segment can indicate the missing TCP segments and the RTO timer does not expire. As a consequence, TCP halves the cwnd and in this way resolves the congestion.

In case C, congestion causes only RLC retransmissions. At 10 sec, TCP linear increase slightly overloads the TN causing Data Frame loss. This causes many RLC PDU retransmissions because a Data Frame contains more RLC PDUs. In this case, retransmissions are successful, thus RLC can handle loss, therefore reset does not occur. It only causes a bit increased delay on TCP level.

As a summary, case C is not a serious problem, but it increases packet delay as well as case B. In the worst case, Data Frame delay is equivalent to TN buffer size, because TCP always fills the buffer, since it controls congestion based on loss. Frequent occurrences of case B and A results in degradation of end-user performance. In case A, TCP transmission stops for a while. This causes unfairness and underutilization at a small number of users. If the number of HSDPA users are greater than one these cases occur even more frequently.

3.1.5 Potential congestion control schemes

Three common congestion control schemes are the credit-based, rate-based and window-based congestion control schemes. In order to adjust the actual rate of a rate-based congestion control or the actual window size of a window based congestion control different algorithms are used. The most common ones among them are the Additive Increase Multiplicative Decrease (AIMD), Multiplicative Increase Multiplicative Decrease (MIMD) and Additive Increase Additive Decrease (AIAD). As already discussed in Chapter 2, to achieve fair bandwidth sharing among flows sharing the same bottleneck, Additive Increase Multiplicative Decrease (AIMD) is proposed. Chiu & Jain (1989) show that AIMD guarantees convergence to fairness; all flows converge to an equal share of resources in steady state, where no flows join or leave.

3.1.5.1 Rate-based congestion control

The 3GPP-standard RAN TN congestion control for HSDPA is credit based; in practice, mainly rate-based solutions is used. In many scenarios the rate-based congestion control scheme proves to be a robust solution but it may result in Transport Network underutilization.

For HSDPA, based on User Buffer Size (UBS), Uu related radio scheduler queue information and the output of congestion detection the Congestion Control algorithm in the Node B decides how many MAC-d PDUs can be transmitted from the RNC for a given flow. This is reported to the RNC using the HS-DSCH Capacity Allocation Control Frame (CA). The CA defines maximum MAC-d PDU length, HS-DSCH Credits, HS-DSCH Interval and HS-DSCH Repetition Period values. The MAC-d shaping in RNC ensures that within a given Interval not more than Credits PDUs are sent. Repetition Period defines how many times the Interval and the Credits are repeated. A newly received CA overrides the old one. Setting the Repetition Period unlimited allows to define an allowed shaping rate, e.g. max PDU length 42 octets, Credits 4 and Interval 10ms settings define shaping rate of 134.4 kbps.

The CA can be used in a rate- or credit-based manner for HSDPA. The main advantage of a rate-based usage is that it generates moderate CA load. High CA load should be avoided because it requires high processing power from the nodes. There are practical problems with using the CA in a credit-based manner. It requires more intense signaling, moreover CAs can be lost, therefore we do not know by when all the credits will have been used. A window-based solution would solve this problem but acknowledgment window is not supported by the standard.

3.1.5.2 Window-based congestion control

Using a window-based solution instead of a rate-based solution, Transport Network utilization can be improved thanks to the self-clocking property. This solution is more TCP friendly.

3.1.5.3 Other methods

Originally, the credit-based HSDPA Flow Control was intended to control radio scheduler queues in Node B (Priority Queue, PQ), therefore some later congestion control solutions also use the framework in a credit-based manner. There are also special or hybrid solutions which combine rate- or credit-based solutions with other, e.g. cross-layer methods.

3.1.6 Existing congestion control solutions

The issue of HSDPA Transport Network Congestion Control has already been addressed in the literature several times.

The most common and mainstream solution is to apply a system-specific flow control/congestion control in the Iub Framing Protocol layer by extending the existing flow control introduced by 3GPP with congestion control functionality, see e.g. Nádas et al. (2007). In this way TN congestion is resolved locally below the RLC protocol layer, RLC handles radio link failures and TCP resolves congestion on the non-3GPP part (e.g. Internet) and Core Network only.

Extending TCP to be informed about TN congestions is also a possible way. However, in this case we would solve a local radio access technology issue by a global solution. Explicit Congestion Notification (ECN) field by Ramakrishnan et al. (2001) is used in IP header for lossless TCP congestion control but could also be used for HSPA Transport Network congestion control. Queue management techniques can also be applied in RNC. These global solutions for a local problem may work fine but it may introduce new issues e.g. fairness to be solved.

Bajzik et al. (2006) introduce cross-layer backpressure in the RNC, which is proactively preventing congestions and RLC retransmissions, when the transport network bottleneck buffer is in the RNC. By the calculation of the CA Credits, the algorithm not only takes into account the information provided by Node B but also the state of the AAL2/ATM level queues in the RNC. Thus, they assume that the Iub Transport Network consists of one link and the state of the AAL2 buffer at its input is known. In this way they can perform a pro-active, aggregated congestion control. The scope of the algorithm is to control the load of the low priority AAL2 buffer so that the delay on the AAL2 layer is not exceeding the maximum allowed delay.

Weigle et al. (2005) presented a new congestion detection and reaction mechanism for TCP, based on measurements of One-way Transit Times (OTT) of TCP segments within a TCP connection. OTTs can more accurately estimate the forward-path queuing delay than round-trip times (RTT). If the RTT is increasing, the sender can not find out whether it is caused by congestion on the forward (data) path, congestion on the reverse (ACK) path, or both.

Vulkán & Nagy (2009) propose a HSDPA QoS aware congestion control solution which guarantees fair resource sharing to connections. The method is able to localize congestion by maintaining throughput and delay estimate for each SRNC using a clustering algorithm and to select the connections whose rate should be reduced in case of congestion over the Iur link.

In the following two subsections a more detailed overview of two solutions is given.

3.1.6.1 Aggregated rate-based congestion control

Nádas et al. (2007) propose an aggregated HSDPA flow level congestion control algorithm that operates in the Iub Framing Protocol (FP) layer. Apart from controlling the MAC-hs Priority Queue it also solves the congestion situation on the transport network. The algorithm is a rate-based congestion control solution, which includes a per flow part, per cell and per Node B level aggregated information. The per flow part is responsible for the fast reaction to a congestion situation in Transport Network or long PQ delays. The cell level aggregation estimates the frequency of scheduling a PQ in the given cell. This is used for estimating the air interface bitrate of the PQ. The Node B (or Transport Network) level aggregation approximates the available Transport Network capacity for HSDPA and distributes it among the flows. The method performs well independently of the Transport Network topology.

3.1.6.2 Per-flow rate-based congestion control

The per-flow rate-based HSDPA Transport Network congestion control by Lundh et al. (2008) has already been presented in Section 1.2.3.2. In order to increase readability, it is reviewed again in this section.

Lundh et al. (2008) propose a HSDPA Transport Network Congestion Control which operates in a per-flow basis, i.e., each HSDPA flow has its own congestion detection, bitrate calculation and shaper part (see Figure 3.1). The main tasks of these three parts are as follows.

The *congestion detection part* in the base station (Node B), aims at finding out the congestion level of the Transport Network based on packets arriving from the Radio Network Controller (RNC) and on the state of the Node B buffers (referred to as MAC-hs Priority Queue in 3GPP (2008a)). Standard 3GPP congestion detection functionalities are used like by Nádas et al. (2007). If Transport Network congestion is detected the bitrate calculation part will be informed.

The *bitrate calculation part* in the Node B calculates the bitrate that is allowed on the Transport Network for the given flow. The congestion control maintains an internal variable for the maximum bitrate of the flow. This bitrate is increased linearly if there is no Transport Network congestion (no reported congestion from congestion detection part). If congestion is reported, the bitrate is reduced multiplicatively. The reduction rate depends on the type of congestion. When a new flow arrives – and therefore a new congestion control entity is created – a Slow Start-like mechanism is used to find out the proper starting bitrate of the flow. It means that the rate is exponentially increased during each round round trip time interval until congestion occurs. After the first congestion the congestion control behaves the above described AIMD manner. The bitrate is calculated per 100 ms. This value is a compromise

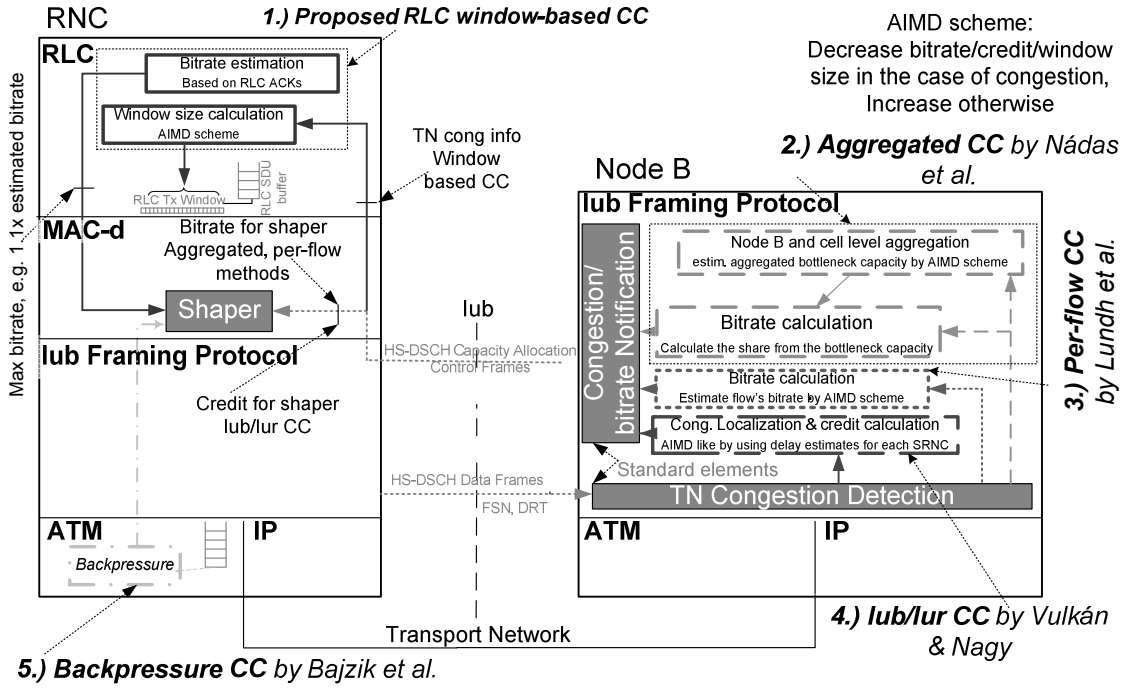

Figure 3.3: Comparison of the proposed method with alternative solutions

among fast reaction, low frequency of Capacity Allocation (CA) control frame sending and low calculation complexity, see Nádas et al. (2007). If the calculated bitrate of the flow changes, then the shaper will be informed about the new bitrate through the Capacity Allocation control frame.

The *shaper* in the RNC shapes the flow according to the signaled maximum flow bitrate. This bitrate is coming from the latest received Capacity Allocation control frame.

3.1.6.3 Comparison of different solutions

In this section, a comparison of four existing HSDPA transport network congestion control methods; the Aggregated Congestion control by Nádas et al. (2007), the Per-flow Congestion Control by Lundh et al. (2008), the Backpressure Congestion Control by Bajzik et al. (2006), the Iub/Iur Congestion Control by Vulkán & Nagy (2009), and the proposed method is given in terms of working principle.

Figure 3.3 shows the main architectural elements of each method. In the figure, the standardized elements, entities and messages, such as the MAC-d Shaper, the Congestion/bitrate Notification, Transport Network Congestion Detection, and HS-DSCH frames are drawn in light gray.

The proposed method (denoted by 1. in Figure 3.3) is an extension of the RLC protocol with congestion control functionality and it relies on the 3GPP congestion detection and signalling framework at Iub Framing Protocol layer. Therefore, it is a

cross-layer solution. Modifications are only needed in the RNC and the solution is still standard compliant. The congestion control follows the AIMD scheme for the RLC transmission window size adjustment. This transmission window determines the average allowed bitrate of the flow. Due to the RLC-specific acknowledgment mechanism, traffic can become bursty. In order to perform traffic smoothing, the MAC-d shaper is also used. The actual shaping rate, which limits the allowed peak rate of the flow, is in proportion with the average flow bitrate estimation to minimize average throughput limitation.

The Aggregated congestion control (2.) by Nádas et al. (2007) makes Node B level and cell-level aggregations in order to estimate the total bottleneck link capacity using the AIMD scheme. This estimated bandwidth is then distributed among the individual flows. The Per-flow congestion control (3.) by Lundh et al. (2008) estimates the flow bitrate by AIMD. The Iub/Iur congestion control (4.) by Vulkán & Nagy (2009) calculates credits and localizes congestion by maintaining throughput and delay estimate for each SRNC using a clustering algorithm. All of these (1.–4.) methods along with the proposed method use the standardized TN congestion detection and notification framework and the MAC-d shaper. The cross-layer backpressure by Bajzik et al. (2006) sends a control command to the MAC-d layer that contains the maximum allowed aggregated send rate of the MAC-d flows if specified thresholds are exceeded at the ATM buffer.

3.2 Proposed method

We propose a window-based congestion control solution which controls the RLC transmission window and the MAC-d shaper according to the congestion level, with the appropriate transmission window size and shaping rate adjustment, and fits in the 3GPP architecture.. From the 3GPP architecture we use the congestion detection functionality and the Shaper part. The proposed solution is basically an AIMD-based congestion control. Radio link failures are handled by the retransmission mechanism of RLC transparently.

The architecture of the proposed RLC-based Transport Network Congestion Control method is shown in Figure 3.4. The congestion control part is implemented in the RLC protocol on the RNC side. In the Node B, the protocol architecture and functionality is similar to that of the 3GPP standard as depicted in Figure 3.1. The TN Congestion Detection part is the same as already described in Section 1.2.3 and Section 1.2.3.2. However, instead of Bitrate Calculation, only Congestion Notification is used. The HS-DSCH Capacity Allocation Control Frames (CA) are not forwarded to the Shaper entity directly as in the case of the standardized framework but they

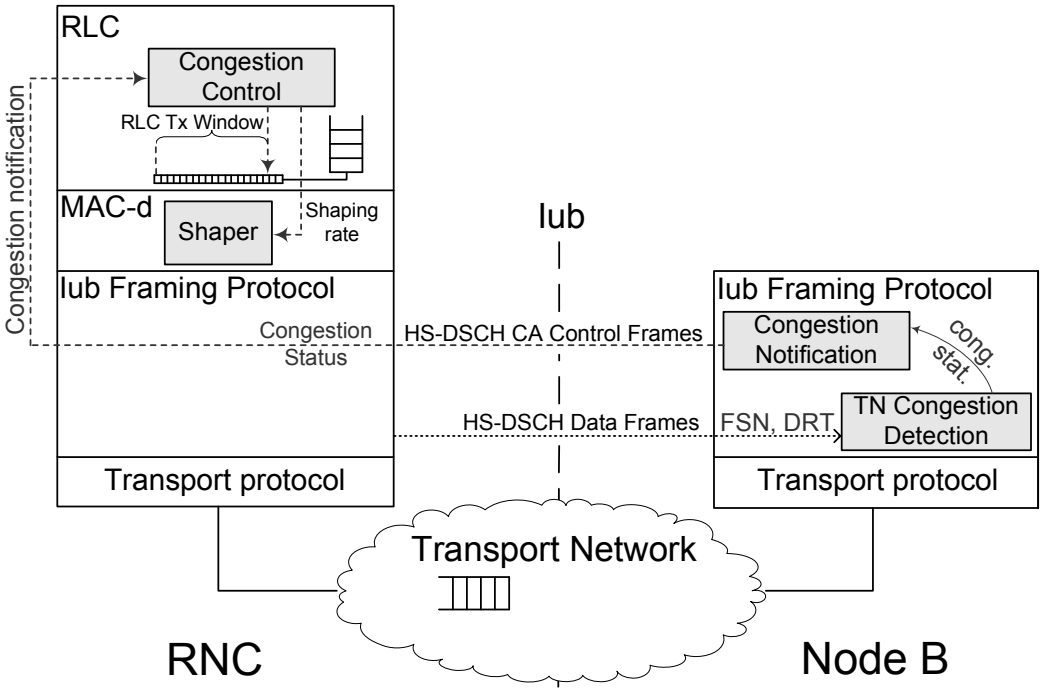

Figure 3.4: RLC-based congestion control architecture

are relayed to the extended RLC entity for congestion avoidance purposes.

The Congestion Control sets the downlink-side RLC Tx window. In HSDPA with the congestion detection method described in Section 3.1.3, all flows perceive almost the same congestion signals from the state of Transport Network, thus the AIMD control scheme leads to an efficient TN usage, an equal share of the TN capacity (fairness), and fast convergence to equilibrium (see Chiu & Jain (1989)). The AIMD scheme does not define the initial window size of a flow. The proposed method uses Exponential Start, with low initial Tx window size (W_{init}=60 PDUs in the examples), until the first congestion if there are no other ongoing flows. *When there are already ongoing flows in the system, the initial window size of a new arriving flow is determined by the Fairness-optimal method [C2], see also Section 2.3. To use this Fairness-optimal method, the actual window sizes of ongoing flows should be known. This is available in the RNC so the Fairness-optimal initial transmission window size is*

$$V^* = \frac{\sum_{i=1}^{n} W_i}{n} + \frac{\frac{\sum_{i=1}^{n} W_i^2}{n} - \left(\frac{\sum_{i=1}^{n} W_i}{n}\right)^2}{\frac{\sum_{i=1}^{n} W_i}{n}},$$

where W_i denotes the actual transmission window size of flow i.

In Exponential Start, Tx window is increased for each acknowledged Protocol Data Unit (PDU) in the RLC status PDU received. After the first detected congestion (turn into Congestion Avoidance state), Tx window is increased with one PDU in each round trip (100 ms), i.e., when a whole Tx window has been acknowledged.

The sketch of this method is shown as Algorithm 2.

After receiving a Capacity Allocation Control Frame in the RNC with Congestion Status *Loss* or *Delay*, the Tx window is reduced. If the Congestion Status in the CA was *Loss*, the Tx window size is reduced by $R_{congestion}$, e.g. 30%. Otherwise, if it was *Delay* and the detected average delay is higher than D_L, e.g. 20 ms, the Tx window size is reduced by R_{delay}, e.g. 10%. If it is even higher than D_H, e.g. 30 ms, the Tx window size is reduced by $R_{congestion}$, e.g. 30%. This delay-based congestion detection scheme performs well in the case of a long TN buffer (e.g., 200 ms). In this case, the method assures that the TN delay is not higher than the target TN delay (~ 40 ms) and loss events are also prevented because of the long TN buffer.

In order to avoid too frequent congestion actions two prohibit timers are started after the last congestion action. All congestion notifications are ignored while the Congestion Detection Prohibit Timer is running. Afterwards, only the most serious congestion event is kept track of, and no congestion action is made while the Congestion Action Prohibit Timer is running. After this timer has expired the Congestion Control reacts to the most serious congestion event and both timers are restarted.

In order to reduce the effect of the problem discussed in Section 3.1.3, the proposed solution is to perform traffic smoothing (to spread bursts) by additional shaping with the use of the MAC-d shaper. We propose that the actual shaping rate should be in proportion with the moving average of the RLC-level throughput estimation described in Section 3.2.1. This shaping, i.e. smoothing of flows in the RNC results in less bursty traffic, which means that packets belonging to different flows are mixed better with each other. In this way the congestion detection will be more fair and also the loss rate will be significantly smaller at the cost of a small additional delay. In addition to this, it does not violate the self-clocking property of the window-based RLC congestion control, see details in Section 3.3.1.

The advantages of the proposed solution are as follows. With the use of 3GPP Congestion Detection framework, only loss on the Transport Network and not on the radio interface is detected. Otherwise, using information only from the RLC protocol, we could not distinguish on which part (on the TN or on the radio interface) of the RLC control loop the congestion occurred, because RLC is terminated in the UE but the Framing Protocol in the Node B. Additionally, in a delay-based RLC congestion detection, uplink delay would be also involved but we only want to focus on downlink. Moreover, using 3GPP Congestion Detection, only the RTT of the TN counts in the control loop and not that of the radio interface. The proposed congestion control is window-based, therefore it has the advantageous self-clocking property. We not only reuse the 3GPP Congestion Detection and shaper functionality but also the standardized signalling framework.

As a summary, the Transport Network Congestion Detection, the Congestion

Notification (Congestion Status field in the CA frame) and the Shaper parts are the same as in the standard. However, bitrate calculation is not performed in the Node B; congestion notification is sent to the new congestion control in the RNC, which adjusts the RLC Tx window and informs the Shaper about the new shaping rate, see Figure 3.1 and Figure 3.4.

```
//Variable tx[i]: actual RLC Tx window size of flow i in PDUs
//Regular update of the Tx window sizes of ongoing flows
for each received CA
    // based on information in the received CA
        if CA has Status Loss or Delay
            then
                decrease tx[i]
                state[i] = Congestion Avoidance
        else
            if state == CongestionAvoidance
                then
                    tx[i]+=(Nr of ACKs in received CA)/tx[i]
            if state == ExponentialStart
                then
                    tx[i]+=(Nr of ACKs in received CA)
end for each received CA

//Updating MAC-d shaping rate
for each ACKed RLC PDU calculate RTT
if Nr of ACKed RLC PDUs  > tx[i]
    then
        estimate AVG(RTT) based on per PDU RTTs
        estimate AVG(RLC bandwidth) based on AVG(RTT)
        set shaping rate to AVG(RLC bandwidth) * F_shaper

//At new flow arrival
Input: tx[] for all ongoing flows
    if tx[] not empty //if there are ongoing flows; calc. init. window
      then tx[new flow]=AVG(tx[])+VAR(tx[])/AVG(tx[]) //of new flow
    else tx[new flow]=W_init then state[new flow] = ExponentialStart
Output: Tx window size of the new flow tx[new flow]
```

Algorithm 2: Sketch of the method implemented in the RNC, see also Figure 3.4

3.2.1 Measurement based estimation of RLC bandwidth

For each RLC PDU, we calculate its RTT (by subtracting from the time of acknowledgement the time it was sent). We take an average value of every $txWindow$ number

of measured RTT samples. For example if the $txWindow$ is 100, then we take the average RTT of the next 100 PDUs. To get an RLC bandwidth estimation, we divide the actual Tx window size (expressed in bit at the end of the measurement period) by the measured average RTT. Note that, this method overestimates the RLC bandwidth when the Tx window is increasing during the measurement period and when the Tx window size is decreasing during the measurement period this is an underestimation. This property is advantageous, because when there is room for bitrate increase and the Tx window is continuously increased then the RLC bitrate is overestimated and in this way the possible limitation of shaping in bitrate increase is further reduced.

3.3 Evaluation of the proposed method

3.3.1 Parameter settings

In this section we review the most important parameters of the RLC-based congestion control and we propose optimal values for them. Considering system limitations we determine the range than we find optimal parameters using simulations. These proposed settings were optimized for 2 to 10 Mbps TN capacity and short (35 to 40 ms) TN buffers. The TN buffer size considered is approximately equal to the target delay of the TN. Therefore, loss-based congestion detection is sufficient. Considering longer TN buffers, delay-based congestion detection is also needed to keep the delay below the target. As an enhancement of the method we also designed and studied delay-based congestion detection.

First, we consider the RLC status sending period T_S. The RLC status-report mechanism is controlled by pre-configured triggers (i.e., timers and/or counters see Section 3.1.2.1) determining the status sending period. This status PDU sending period is important because the maximal achievable RLC level throughput depends on it. Basically, RTT determines the maximal achievable bandwidth, however, if the status sending interval (T_S) is longer than the RTT then the status sending interval determines the maximal achievable bandwidth.

Assuming 656 bit RLC PDU size, 2047 for maximal Tx window size and 40 ms for status interval, the maximum achievable RLC level bandwidth is about 33.6 Mbps.

Basically, frequent status sending would be better because it helps to reduce the number of "stuck" packets which have already been received at the UE but have not been acknowledged yet, i.e., it helps to avoid window stalls. However, too frequent status sending may also worsen RLC retransmission mechanism performance. In this case, unnecessary retransmissions of a PDU may occur before the arrival of the acknowledging status report of that PDU. E.g. if a lost PDU is resent by the Sender

but before it could arrive to the Receiver (due to TN delay), the Receiver sends the negative acknowledgement for that PDU again resulting in a new retransmission of the same PDU while an other copy of it is still on the way. These unnecessarily retransmitted PDUs have to be discarded by the Receiver, therefore they increase the RLC loss rate and decrease TN utilization. On the other hand, if status is reported too rarely, the resulting window stalls cause bad utilization. In order to manage this tradeoff between utilization and loss, T_S should not be set much less than the target RTT of UTRAN between RNC and UE. As a rule of thumb we consider the target downlink delay plus typical average uplink delay to make up this time interval, i.e. $T_S := \mathrm{RTT}_{\mathrm{UTRAN}}^{(\mathrm{target})} \approx T_{\mathrm{DL}}^{(\mathrm{target})} + T_{\mathrm{UL}}^{(\mathrm{typical})}$.

Next, we take a look at the shaper. If the shaper in the RNC is not used (or used at a constant maximal rate), the RLC-based congestion control would not provide satisfying performance results, see Figure 3.5. High loss rate and poor throughput performance would occur. When an RLC status report arrives, arriving acknowledgements trigger the transmission of new PDUs. The Framing Protocol gets these PDUs immediately, wraps them into Iub Data Frames and puts them in the TN. As a consequence, data is put in the TN in bursts. Bursty traffic makes congestion detection unfair and also increases the experienced loss. (Loss rate up to 10–50% may also occur).

In Figure 3.5, it is shown how the RLC Tx window is changing during congestion control operation without using MAC-d shaper. It can be noticed that the first congestion occurs very early, where the Tx window size is still 130 PDUs. The consequences of bursty traffic can also be noticed in terms of unfair congestion detection. The second flow (gray) detects more congestions and instead of convergence of flow rates (Tx window sizes) divergence occurs. Losses occur even if the Tx window is very small (from 33 sec).

To reduce burstiness, shaping should be applied. For the inter-acknowledgement intervals, frames are sent more uniformly distributed if the shaper is properly used. We propose that the actual shaping rate should be in proportion with the moving average of the RLC-level throughput estimation described in Section 3.2.1. We take the estimation of average throughput over RLC RTT periods and we smooth out these samples with a moving average of parameter 0.1. The shaping rate should be higher than this average to avoid limiting the increase of RLC Tx window. However, it should not be too high because in that case bursts may still remain in the traffic. We use the smallest value that already ensures acceptable increase rate; we propose the shaping rate to be F_{Shaper} (e.g. 1.1) times the mentioned average estimated RLC throughput. To illustrate the benefits of the proposed parameter we made a simulative evaluation for the case where the shaper was not used (Fig 3.5) and an other case where the shaper parameter F_{Shaper} was 1.1 (Figure 3.6). In both cases, three users

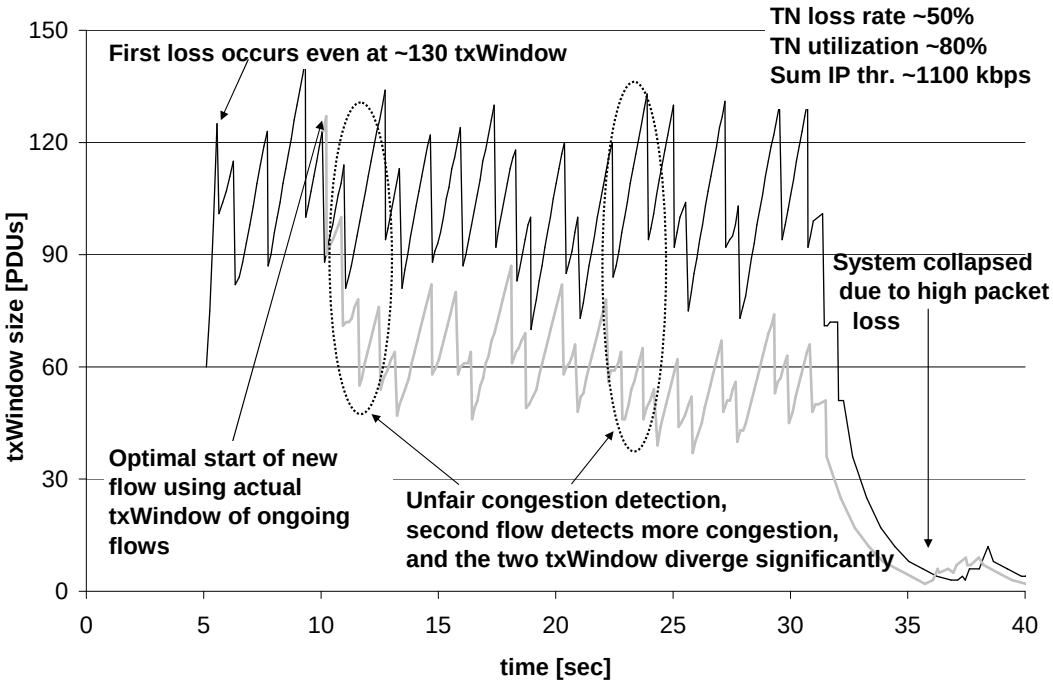

Figure 3.5: Illustration the main drawbacks of the lack of shaping; RLC-based congestion control without shaper, 2 users, file size 100 MByte, TN capacity 2 Mbps, TN buffer length 70 kbit (35 ms), RLC PDU size 656 bits, status sending period 40 ms

downloaded 100 MByte files at 2 Mbps TN capacity, $R_{congestion}$= 30% window size reduction in the case of congestion and status sending period T_S = 40 ms. With the shaper switched on, the achieved aggregated (i.e., summarized for all users) TCP throughput (∼1700 kbps) was one and a half times as high as in the switched-off case (∼1100 kbps). Note that the theoretical maximum TCP throughput is 1794 kbps due to protocol overhead: TCP/IP: 1500/1460, RLC: 84/82, Iub FP: 611/588, UDP/IP: 1500/1470. Frame loss in TN was 50% without shaper and 1% with shaper. The difference in fairness can be seen on RLC Tx window level too. If shaper is applied, flows normally experience congestion in the same time, see Figure 3.6. In this figure the same scenario was simulated as in the case of Figure 3.5, except for the shaper that was not used at Figure 3.5 and used at Figure 3.6. In this scenario, the first loss occurs later, only when the Tx window size is above 280 PDUs. In this case, some unfairness in congestion detection can still be noticed, but the two Tx window sizes converge. If shaper is not used, flows rarely experience congestion simultaneously.

Assuming 40 ms as RLC status sending interval (T_S), RLC Tx window can be doubled in Slow Start in every 40 ms, theoretically. However, assuming 1.1 as shaping parameter (F_{Shaper}) and 100 ms shaper update times, the shaper is limiting throughput, so that changes in the Tx window size takes effect only after 100 ms. Between two shaper updates only 10 % effective increase is possible. The initial (before RLC throughput is available) shaping rate is 100 kbps.

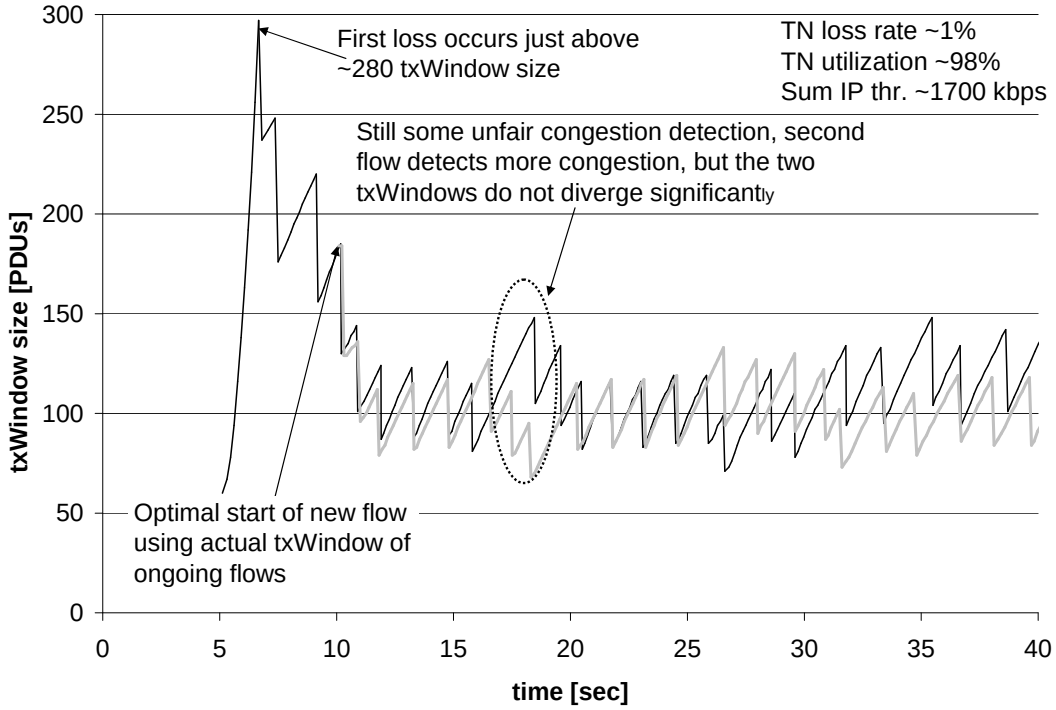

Figure 3.6: Illustration of the operation of the RLC-based congestion control; RLC-based congestion control with shaper, 2 users, file size 100 MByte, TN capacity 2 Mbps, TN buffer length 70 kbit (35 ms), status sending period 40 ms

Without shaper, congestion detection is unfair, the loss rate is high and TCP utilization is poor. Using shaper in the RNC to smooth out packet bursts, the RLC based congestion control provides good performance.

3.3.2 Proposed parameter settings

In order to choose optimal parameter settings we analyzed a large number of simulation scenarios. We examined 17 000 different simulation scenarios, number of users was between 1 and 10, TN link capacity between 2 Mbps and 10 Mbps. The range from which the proposed parameter values have been chosen were determined based on engineering practice taking into account the considerations of the previous section. For the examinations, we used a free data analysis tool called Weka, see Hall et al. (2009).

We first examined the correlation between the two most important key performance indicators (TCP throughput, fairness) and the most important system input parameters, see Table 3.1. These key performance indicators depends mostly on the shaping parameter, that is why we considered this parameter first. By means of the algorithm for finding apriori association rules, see (Jiawei Han (2006)), we analyzed the relations in the system under study. The association rules found showed that throughput and fairness were the lowest when shaper was not used. We got the best

	Congestion factor	Shaper parameter	Status period	Congestion prohibit
TCP thr.	2.49%	24.30%	-8.97%	-0.71%
TCP fairness	1.28%	40.55%	-3.45%	-0.37%

Table 3.1: Correlation of aggregated TCP throughput and fairness with input parameters

results when shaper was used with the parameter F_{Shaper}=1.1 (Other considered values apart from switched-off shaper, F_{Shaper}: 1.2, 1.5). We also used this algorithm to determine the best setting for the status sending period (considered values for T_S: 40 ms, 70 ms, 100 ms.). We only considered scenarios where the shaper was used with parameter F_{Shaper}=1.1. It was found that best fairness can be achieved by the 40 ms status sending interval (T_S), in the case of most of the scenarios independently of the capacity and number of users. We determined the optimal setting of the congestion factor by means of the RepTree (Jiawei Han (2006)) decision tree. The tree shows that $R_{congestion}$=30% reduction in the case of congestion provides the best TCP throughput almost independently of the number of users and link capacity. The proposed setting for delay-based reduction R_{delay}=10% was determined using the same method. Other considered values for $R_{congestion}$ and R_{delay}: 10%, 20%, 30%, 40% . The value of the congestion prohibit timers (Table 3.1) is not in significant correlation with the TCP throughput or fairness, therefore it did not require further analysis in terms of optimization.

As a consequence, the method should be used along with shaper ($F_{Shaper} = 1.1$), because without shaper the fairness and throughput proves to be low. The proposed status sending interval (T_S) is 40 ms, since using 70 ms or 100 ms TCP throughput is lower mainly in the case of a small number of users. The proposed congestion factor $R_{congestion}$ is 30%. The proposed congestion detection prohibit time is 400 ms, and the proposed congestion action prohibit time is 600 ms.

3.3.3 Performance evaluation by simulator

In order to evaluate the performance of the proposed RLC-based TN Congestion Control, the proposed method has been implemented in a 3G/HSPA protocol simulator. The simulator contains HSDPA related protocol functions, such as TCP/IP, Radio Link Control (RLC), Medium Access Control – dedicated (MAC-d), High-Speed Downlink Shared Channel (HS-DSCH) Iub framing and Medium Access Control – high-speed (MAC-hs). The IP-based TN is modelled as a link of fixed capacity with a finite buffer, and fixed downlink propagation delay.

In order to show that the proposed method performs well not only in individual cases and to compare the performance of the original purely loss-based method (for short buffer) and the enhanced loss- and delay based method (for long buffer), several scenarios have been investigated. In all scenarios, RLC status sending period (T_S) is 40 ms, shaper is used. $R_{congestion}$=30% window size reduction is applied in the case of a loss event or longer than 30 ms average dynamic delay (only in the case of long TN buffer). If the average dynamic delay is longer than $D_L = 20$ ms, R_{delay} =10% window size reduction is applied (only with long TN buffer). Size of files to be downloaded is 100 MB, congestion detection prohibit time 400 ms, congestion action prohibit time 600 ms.

The key performance indicators, such as TCP throughput, fairness, and loss of these scenarios (for different number of users and TN capacity) are illustrated in Fig 3.7, Fig 3.8 and Fig 3.9, respectively. In all three figures, on the horizontal axis, number of users (labeled directly below the bars) and the TN link capacity (labeled below the number of users) is shown. In these figures, we considered two different TN buffer lengths. In all four figures, the first bar (for each user number and capacity) shows the result of the appropriate scenario using 35-ms TN buffer size, and the second bar shows that using 200 ms TN buffer size.

In Figure 3.7, TCP throughput performance of the RLC-based Congestion Control is shown. It can be noticed that the combined loss- and delay-based congestion control (in the case of 200 ms TN buffer) slightly outperforms the simple delay-based one (in the case of 35 ms TN buffer) in all scenarios. The reason for that is the fact that increased delay is a good indicator of built-up TN buffers. Therefore the resulting loss events can be prevented if we use longer TN buffers and delay-based congestion detection. As a consequence, slightly higher throughput performance can be achieved.

In Figure 3.8, Jain's fairness index (Jain et al. (1984)) of resource sharing among flows on TCP level is shown. There is no significant difference between the two methods in terms of fairness. However, loss-based congestion detection proves to be slightly fairer in some cases. It is because of the unfairness of the delay-based congestion detection mentioned in Section 3.1.3.

In Figure 3.9, loss rate on the TN is shown. A congestion control using loss-based congestion detection obviously cannot prevent loss events, therefore the TN loss rate is slightly higher if a large number of users share a low-capacity link. The use of delay-based congestion control with long TN buffer is successful at preventing loss events; that is why in Fig 3.9 no black columns can be seen.

As a conclusion, both the purely loss-based and the combined loss- and delay-based solutions provide good throughput performance and fair resource sharing. Since the delay-based solution can prevent loss events, the combined solution for long TN buffers provides even higher throughput at the cost of a small fairness degradation.

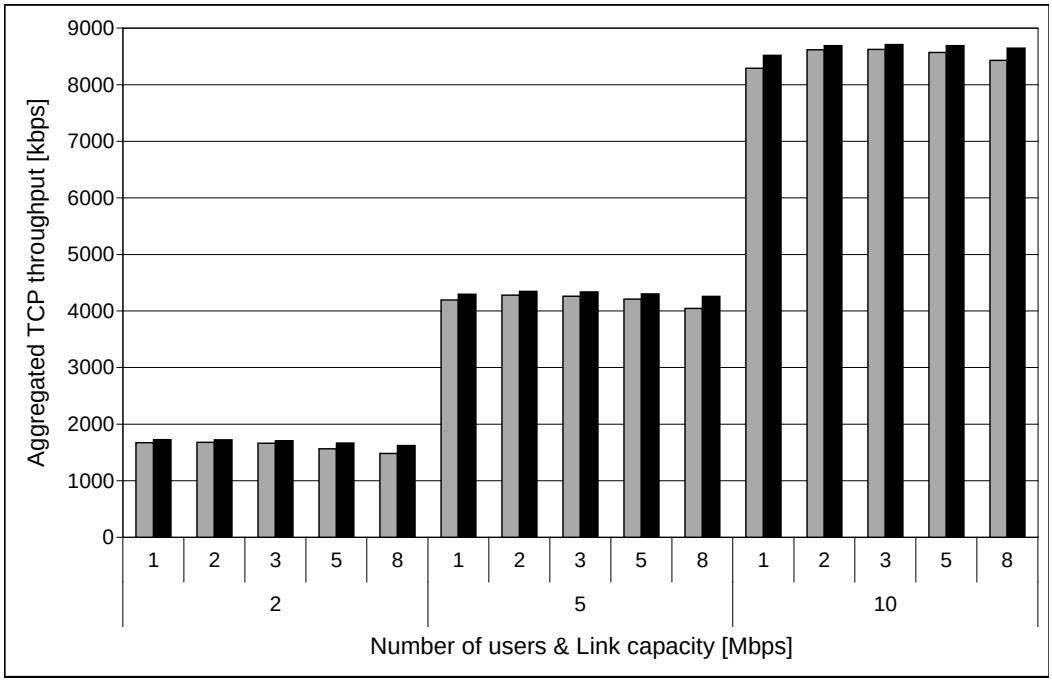

Figure 3.7: Aggregated TCP throughput in the case of short TN buffer (gray) vs. long TN buffer (black)

In the next section it is shown for reference, how HSDPA would work without any congestion control solution.

3.3.4 Comparison with other methods

In this section, four alternative HSDPA TN congestion control solutions are evaluated in terms of performance. The key performance indicators are average TN utilization, loss, fairness, support of different bottlenecks and different buffer lengths.

The aggregated rate-based congestion control solution by Nádas et al. (2007) provides approximately 90% TN link utilization, low TN delay (20 ms) and loss ratio (below 1%). Fairness of resource sharing is not considered by the authors but it is supposed to be good based on the nature of the method. However, their method has no support for different bottlenecks and the authors did not consider different TN buffer lengths, only 120 ms.

The per-flow rate-based congestion control solution described in Section 1.2.3.2 provides slightly lower TN link utilization; approximately 85-90% depending on the number of users in the system. This method also provides low TN delay and loss ratio. Different TN buffer lengths (30 ms and 200 ms) and different bottlenecks are supported by the method. However, this method has fairness issues. However, these issues are addressed and can be improved by the Fairness-Optimal method proposed in Section 2.4.1.

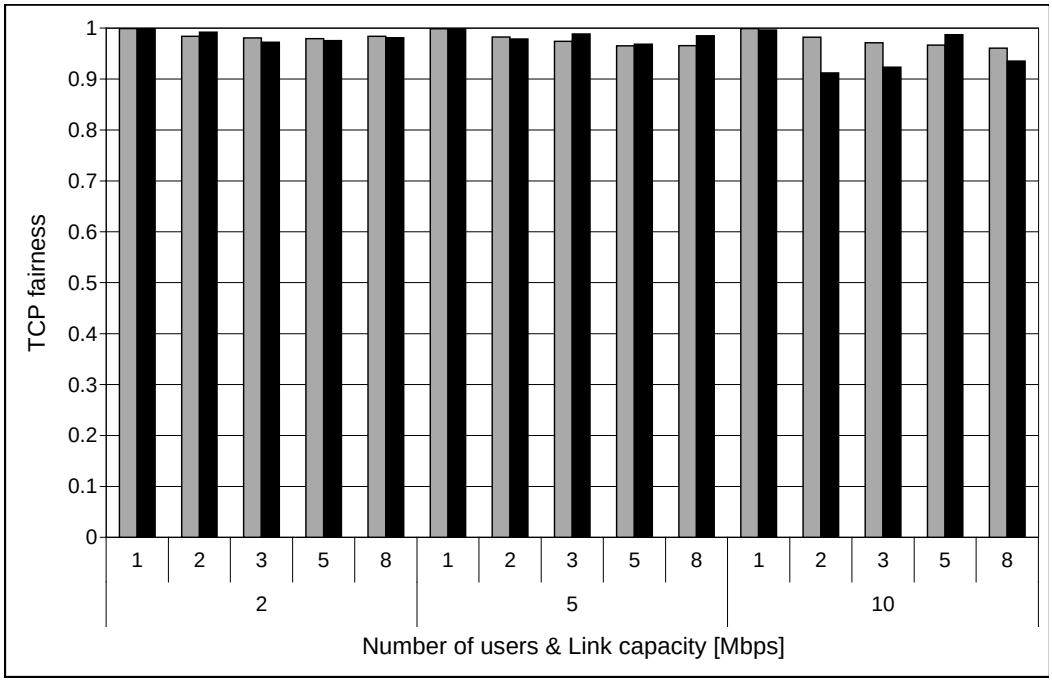

Figure 3.8: TCP level fairness (Jain's fairness index) in the case of short TN buffer (gray) vs. long TN buffer (black)

The cross-layer backpressure method proposed by Bajzik et al. (2006) provides lower TN utilization on average (70–90%), depending on the TN link capacity. The method provides low TN loss ratio and approx. 110 ms TN delay. Fairness of TN resource sharing is not considered by the authors. The method is based on controlling the ATM/AAL2 buffer of the outgoing link at RNC, which is not possible, when the bottleneck buffer is in the Transport Network, but not in the RNC or in the case of a general TN.

The Iub/Iur HSDPA Congestion Control by Vulkán & Nagy (2009) provides high TN link utilization (90–95%), however, it depends on the number of users in the system. The method guarantees fair resource sharing and provides 109 ms mean TN delay.

As a summary, the proposed window-based method provides very high (96%) TN utilization compared to the above presented four methods. Moreover, this high utilization is performed in a wide range of scenarios; it is independent of the number of users or the TN link capacity unlike in the case of the methods proposed by Bajzik et al. (2006) and Vulkán & Nagy (2009). At the same time, the proposed method provides low delay, loss and fair resource sharing. Moreover, this fair resource sharing is also performed among flows with different RTTs (e.g. local server vs. overseas server). The performance comparison of the proposed method with other methods is summarized in Table 3.2.

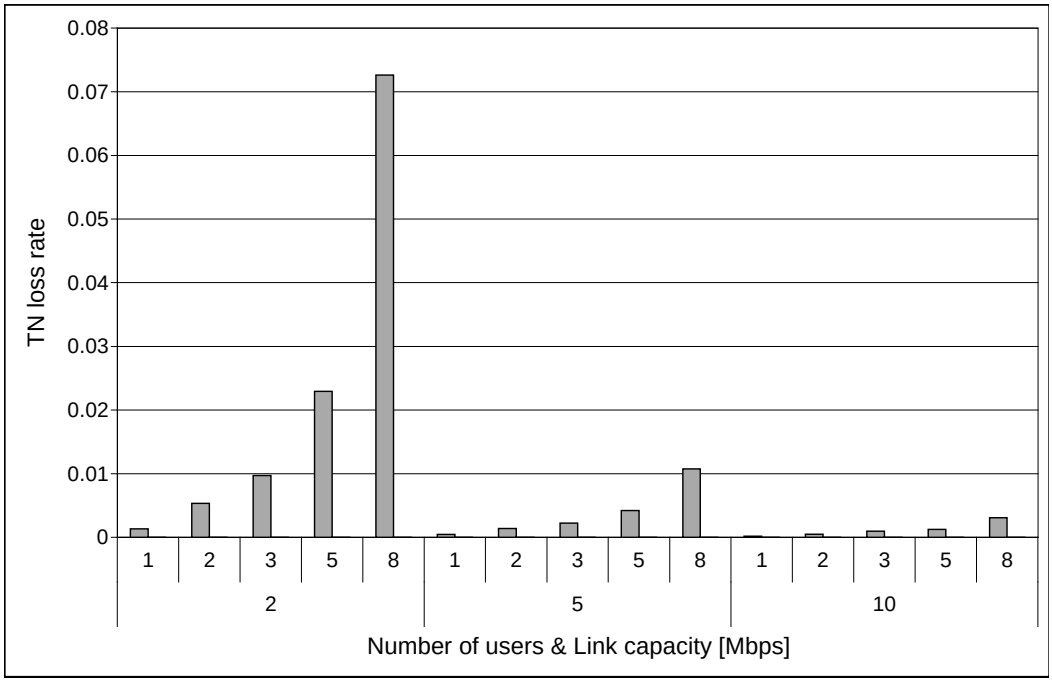

Figure 3.9: TN loss rate in the case of short TN buffer (gray) vs. long TN buffer (black)

3.4 Conclusion

An enhanced, non-standardized, cross-layer window-based HSDPA Transport Network Congestion Control has been proposed. This solution is based on extending the RLC protocol with congestion control functionality and uses the standardized congestion detection and signalling framework along with the standardized shaper. Investigations have shown that the application of a shaper in the RNC to smooth out packet bursts is crucial for getting good performance. In addition to this the proper setting of RLC status sending period is also important. The simulation results support that using the 3GPP shaper is needed for good performance in terms of throughput, fairness, Transport Network loss rate and TN utilization. *It has been shown that the proposed RLC-based method meets the low delay (below 100 ms), loss (∼1%) and high utilization (above 90%) requirements of 3GPP (2001) of a RAN TN Congestion Control.*

Method	Utilization	Fairness	Drawbacks
Nádas et al. (2007)	90%	Good	No support for multiple bottlenecks
Lundh et al. (2008)	85-90%	Medium	Lower fairness, Lower utilization
Bajzik et al. (2006)	70–90%	Medium	Not suitable for general TN, Lower utilization
Vulkán & Nagy (2009)	90–95%	Good	Utilization depends on the number of users
Proposed	96%	Good	Requires RLC protocol stack extension in the RNC

Table 3.2: Performance comparison of the proposed method with other methods

Chapter 4

Peak-rate limited DPS with bandwidth-efficient rate sharing

The previous two chapters discussed congestion control methods. In order to decrease the chance of congestion occurrences in the long run, besides congestion control, link dimensioning is an important issue, as well. The present and the next chapters discuss some open issues of bandwidth sharing models. The research results of these problems, which will be presented in these chapters, are applicable in dimensioning methods, they can also be used in RAN Transport Network dimensioning.

In this chapter, we characterize the state space of the Discriminatory Processor Sharing (DPS) service discipline with peak-rate limitations of the flows.

Discriminatory means that to each traffic class a weight is assigned; and the rate share of the flows are proportional to these weights. Flows from the same class always get the same rate share.

The peak rate limitation means that each traffic class has its own maximal rate. Flows in each class can receive their (maximal) peak rate if there is enough server capacity. Otherwise, some flows will be "compressed" in terms of their actual rate shares compared to their peak rates.

The analyzed rate sharing model is bandwidth-efficient, which means that the server capacity unused by peak-rate limited flows is re-distributed among the non-limited flows.

An efficient algorithmic approach is presented to determine which classes are subject to peak-rate limitations and based on this the bandwidth shares of flows of classes in a given state of this system. Research results related to this chapter are published in [J2]. My theses are related to all the main results of this chapter.

4.1 Introduction

Modern mobile telecommunications networks and high speed data packet services need the elaboration of new resource sharing, congestion avoidance and dimensioning methods, in order to ensure the appropriate Quality of Service (QoS) and Service Differentiation. For link dimensioning (see Jain (1995), [B1]) purposes bandwidth sharing models are needed, especially for the elastic-type (compressible) traffic flows. Such models describing the flow-level performance of elastic flows have been widely studied in the literature, see Koukoulidis (1993), Lassila et al. (2008), Rácz et al. (2002), and Roberts (2004). In this chapter, processor sharing-like models are considered, in which the service capacity (the bandwidth) of the server (the link) is shared among the jobs (flows) according to some sharing principles.

Probably the very first and simplest (egalitarian) processor sharing model was presented by Kleinrock (1967); mainly motivated by the modeling of time-shared computer systems. Fayolle et al. (1980) analyze a single-server processor-sharing system with several classes. Classes are distinguished based on weights, and jobs in the classes have no limitations on their possible service rates (there are no peak-rate limitations) except the server capacity itself. The scheduling strategy considered divides the total capacity in unequal fractions among the different flows according to the corresponding weights; hence such models are called as discriminatory processor sharing (DPS). Fayolle et al. (1980) provide solutions for the conditional expected response time (conditional average waiting time) of a class-k job with a given service time requirement (with a given size of the class-k flow) as well as for the unconditional response times.

Avrachenkov et al. (2005) use the results from Fayolle et al. (1980) and prove that – assuming the system is stable – for each class the expected unconditional response time is finite and that the expected conditional response time has an asymptote.

For multi-class egalitarian processor sharing queues, Cheung et al. (2005) show that the marginal queue length distribution for each class is equal to the queue length distribution of an equivalent single class processor sharing model with a random number of permanent customers. Similarly, the mean sojourn time (conditioned on the initial service requirement) for each class can be obtained by conditioning on the number of permanent customers.

Peak-rate limitations have been introduced and analyzed in a single class processor sharing system first by Lindberger (1999) (called M/G/R Processor Sharing Model) and several improved versions have been studied and proposed for dimensioning IP access networks, e.g., by Riedl et al. (2002) and Fan (2006). Koukoulidis (1993) evaluates multi-rate (peak rate limited) loss models for elastic traffic. A structural characterization of reversibility is developed and used to build a non-egalitarian

processor-sharing queueing discipline that admits a product-form solution. However, in this model a special type of Discriminatory Processor Sharing is considered, where corresponding peak rates and weights are in proportion to each other.

4.1.1 Discriminatory Processor Sharing

Discriminatory Processor Sharing (DPS), see Ayesta & Mandjes (2009), is an important generalization of the (multi-class) egalitarian processor sharing discipline. In DPS, to each traffic class we assign a weight (e.g., service- or priority-class determined by the user's subscription, see Section 4.3 for an example). The weight of class-i is denoted by g_i. The bandwidth share of flows are proportional to these weights. Flows from the same class always get the same bandwidth share. More formally, two requirements can be identified on the capacity shares in DPS:

requirement-A: $\dfrac{c_i}{c_j} = \dfrac{g_i}{g_j}$, and

requirement-B: $\displaystyle\sum_{i=1}^{K} N_i c_i = C,$

where c_i's are the bandwidth shares, K is the number of traffic classes, N_i is the number of class-i users in the system, and C is the server capacity. These requirements are uniquely fulfilled by

$$c_i = \frac{g_i C}{\sum_{j=1}^{K} g_j N_j}, \quad i \in \{1, \ldots, K\}.$$

This is also referred to as the work-conserving property, i.e., either all flows get all the bandwidth they required or the system is serving on its full capacity. This bandwidth share is also the solution of the following optimization problem:

$$\max_{\underline{x}} \sum_{i=1}^{K} N_i g_i \log x_i \quad s.t. \ \sum_{i=1}^{K} N_i x_i = C. \tag{4.1}$$

In this chapter, we characterize the state space and the bandwidth sharing scheme of the peak-rate limited DPS with bandwidth-efficient rate sharing. The peak rate limitation means that each traffic class has its own maximal rate that is denoted by b_i for class-i. If there is enough capacity then the flows receive their peak bandwidths. When there is not enough capacity for all ongoing flows to get their peak rates, that is, $\sum_{i=1}^{K} N_i b_i > C$, then some flows or all flows will be "compressed" in the sense of their reduced service rates. This is the elastic "regime" of the model. On bandwidth-efficient rate sharing it is meant that requirement-B should be fulfilled in the elastic regime of the model; that is, all bandwidth left by the uncompressed flows is to be redistributed among the compressed flows. This type of rate sharing is also referred

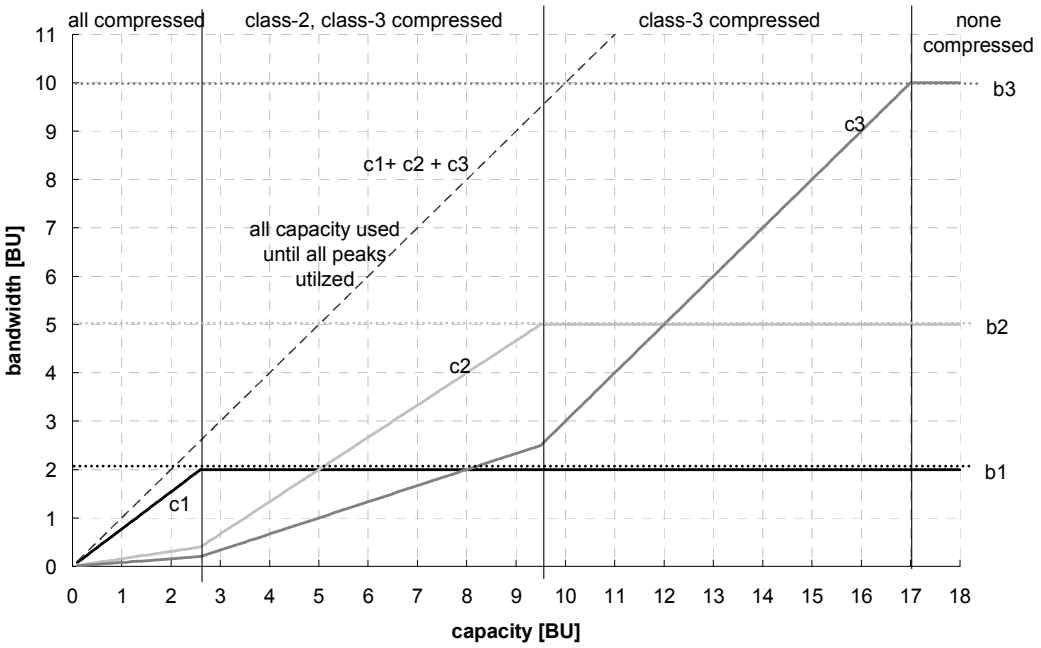

Figure 4.1: Peak-rate limited bandwidth-efficient DPS

	Class-1	Class-2	Class-3
Peak rate (b_i)	2	5	10
Weight (g_i)	10	2	1
Number of users (N_i)	1	1	1

Table 4.1: Parameter settings

to as Pareto-efficient in the literature, see Ayesta & Mandjes (2009).

The chapter is organized as follows. In the next section a comparison is given to illustrate the differences between the bandwidth-efficient and non bandwidth-efficient approaches. In Section 4.2, we show that the bandwidth redistribution leads to a simpler and well interpretable bandwidth share calculation in the case of compressed flows. In Section 4.2.1 we present that there is a strict order of compression and we give a method for determining the set of compressed classes and the bandwidth shares.

4.1.2 Comparison of bandwidth-efficient and non bandwidth-efficient limited approaches

In this section, a comparison of the peak rate limited bandwidth-efficient and the non bandwidth-efficient approaches is given as an illustration.

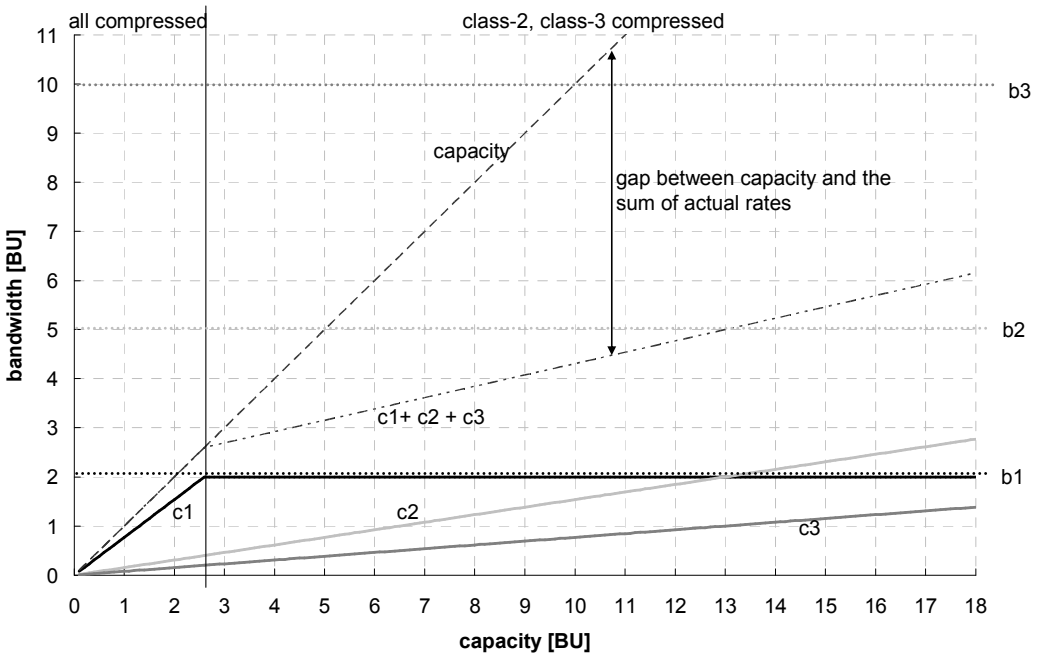

Figure 4.2: Peak-rate limited *non* bandwidth-efficient DPS

Figure 4.1 and Figure 4.2 illustrate how the bandwidth-efficient and non bandwidth-efficient approaches are different from each other. Both figures show the same scenario, the only difference is the bandwidth-efficiency. See class parameters in Table 4.1. On the horizontal axis the capacity is shown. It is increased from 0 to 18 Bandwidth Units. On the vertical axis the actual service rate of the given class (c_i) is plotted. The peak rate of each class (b_i) is also shown.

The main difference between the two approaches is that in the case of the non-bandwidth-efficient approach, the total capacity is fully utilized only if all classes are compressed (Figure 4.2, where capacity is less than 2.7), i.e., not the peak rates are the limiting factors in the service rates. Otherwise, in this case the capacity is not utilized because residual capacity left by the peak-rate limited class 1 is not fully redistributed among non peak-rate limited class 2 and 3. In the case of the bandwidth-efficient approach (Figure 4.1), the available capacity is always fully utilized, except when the service rates of all flows are limited by their peak rates.

In Figure 4.1, four regions can be distinguished. If capacity is less than 2.7, all classes are compressed. This is the only region, where the bandwidth-efficient and the non bandwidth-efficient approaches give the same service rates, because there is no unused capacity left from peak-rate limited classes. If the capacity is not less than 2.7 but less than 9.5, then class-1 is no longer compressed, i.e., it gets its peak rate (b_1). Class-2 and class-3 are still compressed in proportion of their weights. If the capacity is not less than 9.5 but less than 17 then class-2 receives its peak rate also and is no longer compressed. In this region only class-3 is compressed, but gets all

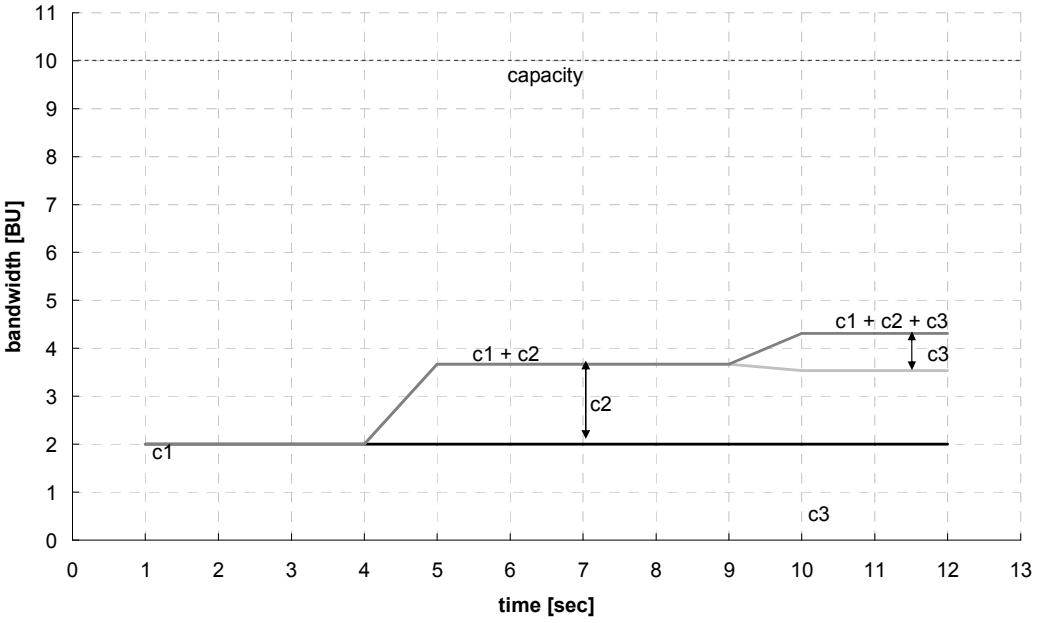

Figure 4.3: Peak-rate limited *non* bandwidth-efficient DPS, capacity=10 BU

the capacity left from both peak-rate limited classes. If the capacity is not less than 17 then all classes receives their peak rates. In this region, because all classes are limited by their peak rates, further increase of the capacity does not increase the sum of the services rates of the three classes.

In Figure 4.2, only two regions can be distinguished. If capacity is less than 2.7, all classes are compressed. This is the only region, where the bandwidth-efficient and the non bandwidth-efficient approaches give the same service rates. Therefore, this region of Figure 4.2 is the same as that of Figure 1. If the capacity is not less than 2.7 then class-1 already gets its peak rate (b_1), class-2 and class-3 are still compressed and their service rate is calculated according to the same formula as in region 1. Since capacity left by the peak rate limited class-1 is not redistributed among class-2 and class-3, there is a gap between the capacity and the sum of the actual service rates. The higher the capacity, the larger this gap gets.

Figure 4.3 and Figure 4.5 give an other illustration of the difference between the bandwidth-efficient and the non-bandwidth efficient approaches. In both figures the same scenario is depicted, apart from the bandwidth-efficiency. See class parameters in Table 4.1. Capacity is now fixed at 10 BU. Figure 4.4 also shows the same scenario but no peak rates are used. In these three figures, time is plotted on the horizontal axis. On the vertical axis the actual service rate of the given class (c_i) is shown in a cumulative way. It means that instead of plotting c_1, c_2, c_3 individually, c_1, the sum of c_1 and c_2, and the sum of c_1, c_2, and c_3 is plotted. At 0 sec no flows are in the system. At 1 sec a flow from class-1 arrives, at 5 sec a flow from class-2 arrives, finally at 10 sec, a flow from class-3 arrives.

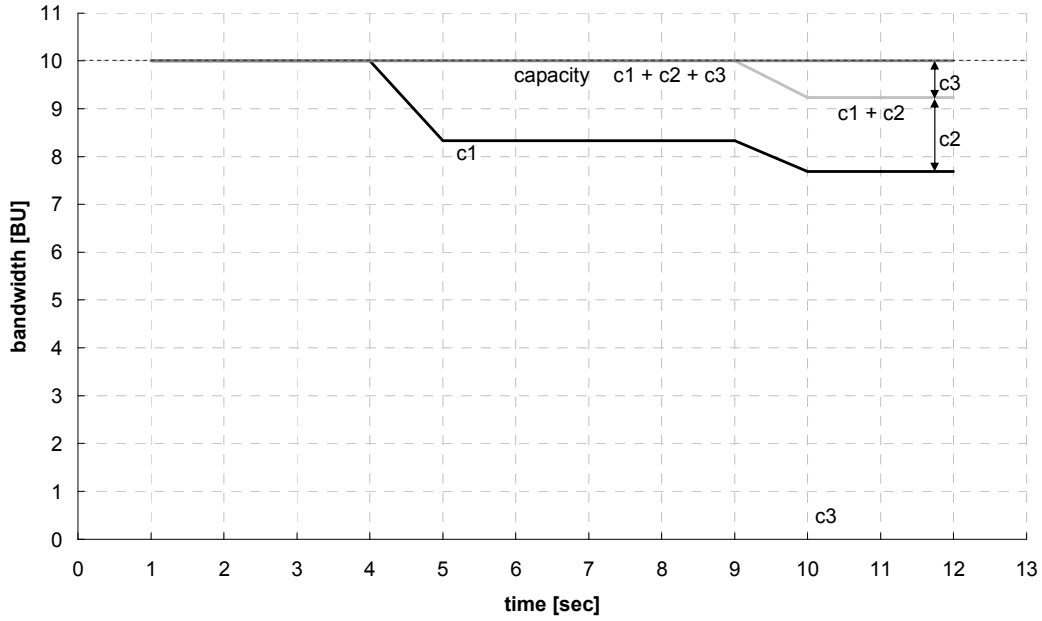

Figure 4.4: DPS with no peak rates, capacity=10 BU

Figure 4.3 shows the peak-rate limited non bandwidth-efficient approach. When class-1 arrives at 1 sec, it gets its peak rate. When class-2 arrives at 5 sec, its service rate is calculated according to (4.2), therefore it cannot utilize its peak rate. The same applies to class-3 when it arrives at 10 sec. The total capacity can still not be utilized.

Figure 4.4 depicts the same scenario with *no* peak rate limitations. When class-1 arrives at 1 sec, it can use the total capacity. When class-2 arrives at 5 sec the two classes share the capacity in proportion of their weights. When class-3 arrives at 10 sec the capacity is shared among three flows in proportion of their weights. The total capacity is always utilized since no peak rates are limiting the flows service rates.

Figure 4.5 shows the peak-rate limited bandwidth-efficient approach. When class 1 arrives at 1 sec, it gets its peak rate. When class-2 arrives at 5 sec it also gets its peak rate since the sum of the peak rates (7 BU) is still less than the capacity (10 BU). When class-3 arrives at 10 sec, it gets compressed since it has the smallest weight and otherwise the capacity would be exceed. The total capacity is only utilized after 10 sec because until this time peak rate is limiting both flows.

4.2 Bandwidth share calculations in peak-rate limited DPS

The non bandwidth-efficient processor sharing has been widely studied in the literature, but it does not prove to be a realistic model for real systems. In a non

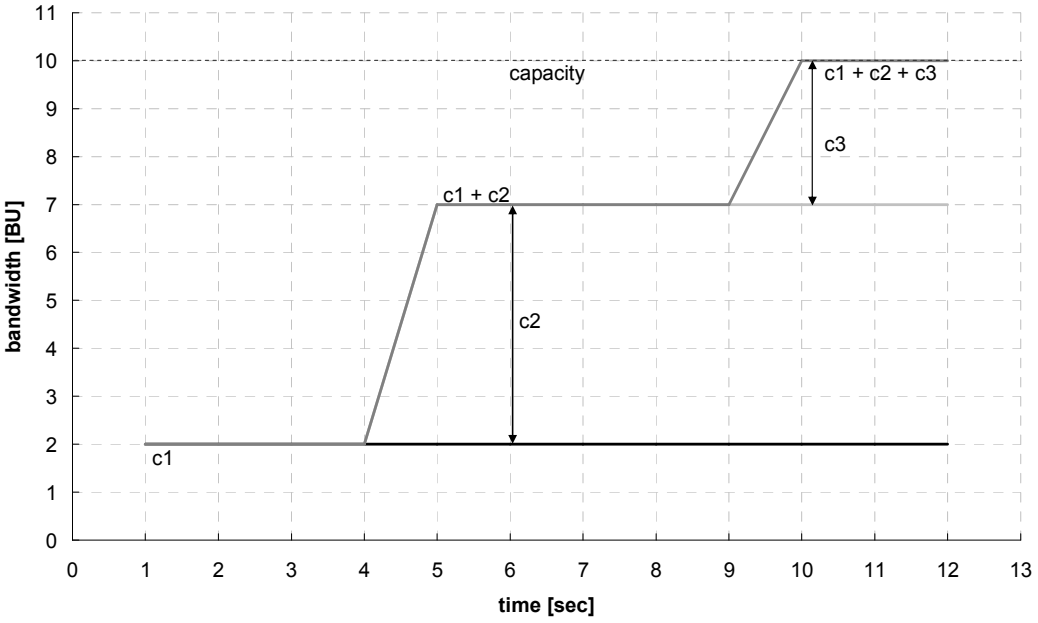

Figure 4.5: Peak-rate limited bandwidth-efficient DPS, capacity=10 BU

bandwidth-efficient case the total available capacity may not be used because residual capacity left by peak-rate limited flows is not (fully) redistributed among non peak-rate limited flows. In the models analyzed by Ayesta & Mandjes (2009) and Lakshmikantha et al. (2010) there is no redistribution at all of unused capacity, hence bandwidth share of flow-i can be simply calculated in the following way:

$$c_i = \min\left(\frac{g_i C}{\sum_{j=1}^{K} g_j N_j}, b_i\right). \tag{4.2}$$

Ayesta & Mandjes (2009) only discuss the non bandwidth-efficient case and consider the bandwidth-efficient case to be harder to analyze.

According to bandwidth-efficient rate sharing, unused capacity is redistributed among flows in proportion to their weights, so the calculation of the bandwidth shares and determining the set of compressed classes are somewhat more complicated. For a while, let us assume that the set of compressed ($\mathcal{Z}$:$\{\forall i, c_i < b_i\}$) and uncompressed ($\mathcal{A}$) classes are known in a given state $\underline{N} = (N_i,\ i \in \{1, \ldots, K\})$. In the uncompressed case, $c_i = b_i,\ i \in \mathcal{A}$. Since these flows cannot utilize their bandwidth shares, they leave

$$\sum_{i \in \mathcal{A}} \left(\frac{g_i N_i}{\sum_{j=1}^{K} g_j N_j} C - N_i b_i \right)$$

capacity which is re-distributed among compressed flows. If $j \in \mathcal{Z}$, the original bandwidth share is increased due to the redistribution. The redistribution should be proportional to the weights g_i, in order to keep a similar requirement to requirement-A. Between two compressed classes, $c_i/c_j = g_i/g_j,\ i, j \in \mathcal{Z}$, and between a compressed

and an uncompressed class, $c_i > c_k \frac{g_i}{g_k}, \forall i \in \mathcal{Z}, \forall k \in \mathcal{A}$. (The latter requirement is needed to ensure that $\mathcal{Z}$ is unique for given $\underline{N}$.) This results

$$c_i = \frac{g_i}{\sum_{j=1}^{K} g_j N_j} C + \frac{g_i}{\sum_{k \in \mathcal{Z}} g_k N_k} \sum_{l \in \mathcal{A}} \left(\frac{g_l N_l}{\sum_{j=1}^{K} g_j N_j} C - N_l b_l \right), i \in \mathcal{Z}. \tag{4.3}$$

Due to our assumption, constraint $c_i \leq b_i$ is fulfilled for $i \in \mathcal{Z}$.This formula shows that identifying the service rate of classes and the set of compressed classes are more complicated in the bandwidth-efficient approach.

For an illustration of the differences between the bandwidth-efficient and the non bandwidth-efficient approaches see Section 4.1.2. In the following, the bandwidth-efficient method is considered.

For implementing a calculation of bandwidth-efficient rate shares based on (4.3) we first show a simpler form of that equation, and then using this simpler form, we present a method for determining $\mathcal{Z}$ and $\mathcal{A}$.

Lemma 4.2.1. *The service rate of the compressed classes' users formulated in* (4.3) *can be re-written as*

$$c_i = \frac{g_i}{\sum_{j \in \mathcal{Z}} g_j N_j} \left(C - \sum_{k \in \mathcal{A}} N_k b_k \right), i \in \mathcal{Z}. \tag{4.4}$$

The proof of this lemma is based on taking the right-hand side of (4.3) over a common denominator and performing a simplification, which eventually results in the right-hand side of (4.4). The immediate consequence is that c_i, $i \in \mathcal{Z}$ can be considered as the bandwidth allocation of a reduced Discriminatory Processor Sharing system with capacity $(C - \sum_{k \in \mathcal{A}} N_k b_k)$ and traffic classes $\mathcal{Z}$ in state $\underline{N}$.

Note that $\mathcal{Z}$ is unique for a given $\underline{N}$. Two cases can be distinguished. If $\sum_{i=1}^{K} N_i b_i \leq C$, $\mathcal{Z}$ is empty so solution $c_i{=}b_i$ evidently fulfills constraint $\sum_{i=1}^{K} N_i c_i \leq C$. In the second case, $\sum_{i=1}^{K} N_i b_i > C$ and hence $\sum_{i=1}^{K} N_i c_i = C$. From this, it follows that there exists a class i^* for which $c_{i^*} < b_{i^*}$, that is, there is at least one compressed class (i^*). If class j is also compressed, $\frac{g_{i^*}}{g_j} = \frac{c_{i^*}}{c_j}$ holds by definition. Thus, $c_j = \min\left(b_j, \frac{g_j}{g_{i^*}} c_{i^*}\right)$, which means that $c_j, \forall j$ can be calculated from the bandwidth share c_{i^*} of one compressed class i^*. As a consequence,

$$\sum_{j=1}^{K} N_j \cdot \min\left(b_j, \frac{g_j}{g_{i^*}} c_{i^*}\right) = C. \tag{4.5}$$

The left-hand side of (4.5) is monotonously increasing function of c_{i^*}, so while $\sum_{i=1}^{K} N_i b_i > C$, there is one solution for c_{i^*}. Therefore, there is one solution for each c_j.

Preliminary numerical calculations lead us to conjecture that the bandwidth allocation from (4.4) is a global solution of the following optimization problem which differs from (4.1) in the constraint $x_i \in [0, b_i]$:

$$\max_{\underline{x}} \sum_{i=1}^{K} N_i g_i \log x_i \;\; s.t. \;\; \sum_{i=1}^{K} N_i x_i \leq C \;\; \& \forall i \; x_i \in [0, b_i].$$

The proof is given by Bíró et al. (2013). The next section illustrates the difference between the peak rate limited bandwidth-efficient and the non bandwidth-efficient approaches. In Section 4.2.1, we present an algorithmic approach to determine the set of compressed classes $\mathcal{Z}$ in the next section.

4.2.1 Determining the compression of classes

In this section, a method is proposed for determining the set of compressed classes $\mathcal{Z}$ and also the bandwidth shares of flows from each class.

Let C denote the considered capacity. Three disjunct cases are distinguished considering the compression of classes; namely all classes are uncompressed, all classes are compressed, and there are compressed classes but at least one class is uncompressed.

Let $\mathcal{S}$be a subset of $\{1,\ldots,K\}$. In the following three steps below, $\mathcal{S}$ will be adjusted. Initially, let $\mathcal{S}$=$\{1,\ldots,K\}$.

Step-1: Check whether all classes are uncompressed. If it is true, then C is enough for peak rates of all flows, i.e.

$$\sum_{i \in \mathcal{S}} N_i b_i \leq C \; . \tag{4.6}$$

In this case every flow gets its peak rate: $c_i = b_i, \; \forall i \in \mathcal{S}$

and all classes are contained by $\mathcal{A}$. No further steps are needed.

Step-2: Check whether all classes are compressed. If it is true, the bandwidth share of all flows are less than their peak rates, and there is no unused capacity to be redistributed, i.e.,

$$\frac{g_i}{\sum_{j \in \mathcal{S}} N_j g_j} C < b_i, \forall i \in \mathcal{S} \; . \tag{4.7}$$

The bandwidth share of flows are

$$c_i = \frac{g_i}{\sum_{j \in \mathcal{S}} N_j g_j} C, \; \forall i \in \mathcal{S},$$

so every class gets bandwidth share in proportion to their weights and all classes are contained by $\mathcal{Z}$. No further steps are needed.

Step-3: In this case there are compressed classes, but at least one class is uncompressed, because none of the conditions in Step-1 (4.6) and Step-2 (4.7) is fulfilled.

To determine which class is surely uncompressed we use the following equivalence

$$\frac{g_i}{b_i} < \frac{\sum_{j\in\mathcal{S}} N_j g_j}{C}, \forall i \in \mathcal{S} \Leftrightarrow \max_{i\in\mathcal{S}} \frac{g_i}{b_i} < \frac{\sum_{j\in\mathcal{S}} N_j g_j}{C} \tag{4.8}$$

The left-hand side of the relation simply comes from (4.7) by rearrangement. The equivalence above also means that if (4.7) is not fulfilled, then the right-hand side of (4.8) is also not fulfilled. Consequently, this surely uncompressed class is i',

$$i' = \arg\max_{i\in\mathcal{S}} \frac{g_i}{b_i}, \text{ with bandwidth share } c_{i'} = b_{i'}.$$

For the remaining classes we should evaluate a reduced system where the effect of this class is considered according to Lemma 4.2.1; that is, the considered capacity is reduced by $N_{i'}b_{i'}$ (i.e., $C \leftarrow C - N_{i'}b_{i'}$) and only the remaining classes are considered ($\mathcal{S} \leftarrow \mathcal{S}\backslash\{i'\}$). For the reduced system we continue with Step-2.
The above described method can be summarized as the following algorithm (Algorithm 3):

1. $\mathcal{Z} = \{1, 2, ..., K\}$
2. *while* $\max_{i\in\mathcal{Z}} \left\{\frac{g_i}{b_i}\right\} \geq \frac{\sum_{j\in\mathcal{Z}} N_j g_j}{C}$ *and* $\mathcal{Z} \neq \emptyset$ *do*
   $i' = \arg\max_{i\in\mathcal{Z}} \left\{\frac{g_i}{b_i}\right\}$
   $\mathcal{Z} \leftarrow \mathcal{Z}\backslash\{i'\}$
   $C \leftarrow C - N_{i'}b_{i'}$
3. *for* $i = 1$ *to* K *do*
   if $i \in \mathcal{Z}$ *then* $c_i = \frac{g_i}{\sum_{j\in\mathcal{Z}} N_j g_j} C$
   else $c_i = b_i$.

Algorithm 3: Determining the set of compressed classes

An important consequence of the above described method is that the following order of classes:

$$\frac{g_1}{b_1} \leq \frac{g_2}{b_2} \leq \cdots \leq \frac{g_K}{b_K},$$

based on the ratios g_i/b_i, is directly related to the compressed and uncompressed classes in such a way that:

if a class with given g/b value is compressed, then all classes with lower g/b are compressed, and if a class with given g/b value is uncompressed, then all classes with higher g/b are uncompressed. Also, observe that the compression order depends on neither the server capacity nor the number of users. In addition to this, the

above described method also provides bandwidth shares of flows in each classes, as formulated in (4.4).

4.3 Applicability of the results

The bandwidth-efficient multi-class discriminatory processor sharing model with limited access rates characterizes adequately the bandwidth sharing policy of more telecommunications systems. A good example for that is the mobile broadband data access technology HSDPA (see 3GPP (2004)). In this system, the access rate limit of each class/user (b_i) comes from the capability (see 3GPP (2011)) of the User Equipment (UE) used. The weight of each class (g_i) may come from the service package the user is subscribed to, e.g., 'gold', 'silver', 'bronze' subscriptions (see Sprenkels et al. (2000)), where 'gold' represents the best service (e.g., highest bandwidth or priority), and 'bronze' the worst. In many practical scenarios the Transport Network represents the bottleneck [B1] capacity C instead of the radio interface (Uu), because the radio part of the network is often enhanced faster than the terrestrial part due to cost-efficiency reasons. It also means that the bandwidth sharing algorithm, which is obviously bandwidth-efficient and considers class weights, not only has to be implemented in the Uu scheduler but in the Transport Network resource sharing control [B1] too. This resource sharing control determines how the Transport Network bottleneck should be shared among users.

A typical parameter setup for the HSDPA system would be for example the following. The Transport Network bottleneck C is a 20 Mbps leased line that connects the base station and the Radio Network Controller (RNC). Users in the 'gold' service class (class-2) are twice as important as users in the 'bronze' class (class-3), and 'silver' class is twice as important as the 'bronze' class. It means that $g_1 = 2$, $g_2 = 3$, and $g_3 = 1$. Let's assume that users in class-2 have UE with 7.2 Mbps maximum data rate capability ($b_2 = 7.2$ Mbps), and users in class-1 and class-3 have UE with $b_1 = 3.6$ Mbps, $b_3 = 3.6$ Mbps. If we have, e.g., 1 'gold' user, 2 'silver' users, and 1 'bronze' user in the system ($n_1 = 2$, $n_2 = 1$, and $n_3 = 1$), the sum of the users' maximum data rates does not exceed the network capacity, so each user can utilize its access rate limit. If one more 'bronze' user arrives to the system, i.e., $n_2 = 1$, $n_1 = 2$, and $n_3 = 2$, capacity is exceeded so the 'bronze' class gets compressed. If, e.g., 3 more 'bronze' users come, i.e., $n_1 = 2$, $n_2 = 1$, and $n_3 = 5$, all classes gets compressed (see also Figure 4.6). In all cases, the bandwidth shares of compressed classes can be calculated according to (4.4).

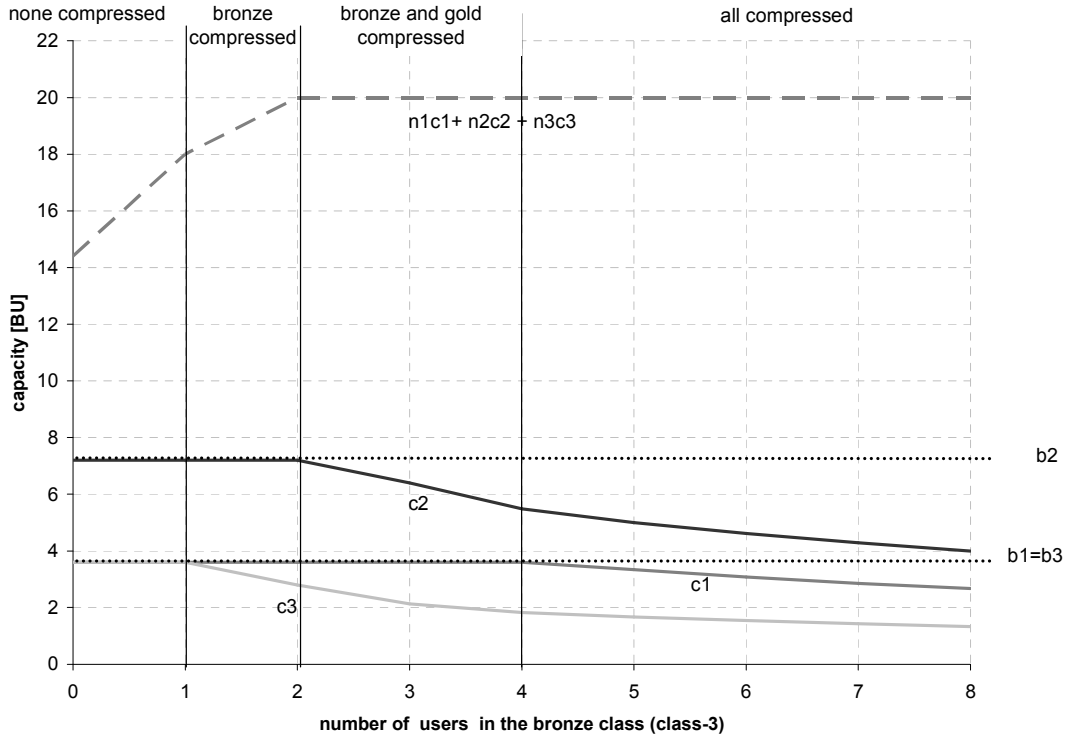

Figure 4.6: Bandwidth sharing among the classes as a function of the number of users in the bronze class (class-3)

4.4 Conclusion

This chapter is concerned with discriminatory processor sharing model with peak-rate limitations and a bandwidth-efficient rate sharing. We have shown that at a given state space (as a snapshot) the peak-rate limited DPS system includes a reduced-capacity DPS system over the compressed classes. Based on this we have given a method to determine the set of compressed and uncompressed classes of flows and their bandwidth shares. It has also been shown that there is a strict order of classes in which they become compressed, and this order coincides with the order of ratios of the corresponding class weights and class peak rates ($\frac{g_i}{b_i}$).

The significance of the result presented in this chapter lies in the fact that these are inevitably the starting points for both the analysis and simulation of Discriminatory Processor Sharing constrained by peak-rate limitations and unused capacity redistribution.

Chapter 5

Characterization of a multi-rate system supporting stream and elastic services

This chapter presents the results of a cooperation in which my main contribution was the elaboration of an efficient method for evaluating the stationary probability of the so-called macro-states and the numerical validation of the proposed method. Hereby, I include the entire work in order to make the results more understandable, but my theses are only related to the evaluation of the stationary probability of macro-states (Section 5.3.3) and the numerical validation part (Section 5.4). Research results related to this chapter are published in [J1].

In this chapter, we calculate the average throughput of elastic flows in a multi-rate (i.e., multi-class) stream-elastic system. Stream flows are characterized by a fixed bandwidth assignment (e.g. Guaranteed Bit Rate, GBR). Elastic flows (e.g. data) and rate-adaptive flows (e.g. video) can adapt their service requirements and share the capacity left over by stream flows. The sojourn time of elastic flows is affected by the assigned service rate whereas the sojourn time of rate-adaptive flows is not affected by that.

Integration of stream, elastic and rate-adaptive traffic has been widely studied in the literature. Fayolle et al. (1980) analyze a single-server processor-sharing system with several classes. Classes are differentiated based on weights and there are no peak rate limitations. The scheduling strategy considered divides the total capacity in unequal fractions among the different flows according to the corresponding weights. The cited paper provides an expression for the conditional expected response times of class k flow with a given required service. In this chapter, *instead of weights, peak rates of traffic classes are modelled.*

Logothetis & Moscholios (2007) give an overview of multi-service loss models.

Models are put into four categories: random traffic, quasi-random traffic, batched poisson traffic and on-off traffic. The classification is based on the flow arrival process, the bandwidth requirements of flows and on the queuing discipline. Our model belongs to the category of random traffic, where flows arrive randomly with either fixed or elastic bandwidth requirements.

Litjens et al. (2008) present an analytical study of throughput measures in processor sharing queuing systems with stream traffic, rate-adaptive and elastic traffic. The authors only consider *one traffic class* per traffic type. They show that the average flow throughput is well approximated by the instantaneous throughput – which is easily obtained from the steady state distribution of the system – for each of the investigated models. In this chapter we also consider the instantaneous throughput as performance measure.

Delcoigne et al. (2004) use an M/G/1 processor sharing model to evaluate the performance of the integration of *one stream and one elastic class* in a multi-service network. The authors consider the expected time to complete the transfer (expected response time) as key performance indicator for elastic flows. The authors assume Poisson flow arrivals, priority service for streaming traffic and fair bandwidth sharing for elastic flows, thus the underlying stochastic system to be an M/G/1 processor sharing queue with time-varying capacity.

Multi-rate loss models with multiple traffic classes, i.e. with different peak rate limitations, have also been studied. The Kaufman-Roberts formula (see Dziong & Roberts (1987), Kaufman (1981)) describes blocking probabilities for *stream* flows in a multi-rate loss model and it is an extension of the Erlang-B formula.

Koukoulidis (1993) evaluates multi-rate *elastic* models. A structural characterization of reversibility is developed and used to build a non-egalitarian processor-sharing queuing discipline that admits a product-form solution. Borst & Hegde (2007) analyze the flow-level performance of a wireless system supporting a combination of multi-rate stream and elastic traffic in terms of transfer delays and user throughputs where stream flows are *rate-adaptive*. Rácz et al. (2002) approximate a system including rate-adaptive stream and elastic traffic classes by a quasi-reversible model, which forms the basis of a recursive approximate solution, similar way as it is done by Koukoulidis (1993) for elastic traffic only.

A fundamental property that results in product-form queueing disciplines is the quasi-reversibility property (Kelly (1979)). The importance of the quasi-reversibility is that it allows constructing efficient recursive formulas. The above mentioned models by Dziong & Roberts (1987), Kaufman (1981), Koukoulidis (1993), Rácz et al. (2002), and Borst & Hegde (2007) are or can be approximated by a quasi-reversible system. The quasi-reversibility is a sufficient criteria for the insensitivity property as well. Insensitivity means that the stationary distribution is insensitive to the service time

distribution, as long as the mean is held fixed, see Burman (1981). Note that Dziong & Roberts (1987) and Kaufman (1981) give exact solutions but Koukoulidis (1993), Rácz et al. (2002), and Borst & Hegde (2007) give approximations.

Our considered system does not lead to a quasi-reversible model and it cannot be approximated accurately with a quasi-reversible system. In this chapter, we trace back the multi-rate stream-elastic model to a two-dimensional one and we solve that directly by determining the steady-state distribution. However, if stream classes are considered only, the model is quasi-reversible and the formula by the Kaufman-Roberts formula (see Dziong & Roberts (1987), Kaufman (1981)) is applicable. If we take only elastic classes into account, the model can be approximated by a quasi-reversible system as shown by Koukoulidis (1993) and Rácz et al. (2002). In this chapter, we propose an efficient numerical approximative method that handles many stream and many elastic classes with different peak rates. There is no restriction on the fraction of mean holding times of elastic flows and stream flows. However, the proposed method provide more accurate results for scenarios where the mean holding times of the elastic flows are the same and the mean holding times of the stream flows are also the same, respectively. The computing complexity of the method does not depend on the number of traffic classes.

A variety of numerical methods that address stream-elastic models (see Logothetis & Moscholios (2007)) has been studied. These methods differ in three ways. First one is the level of detail the model captures, e.g., traffic types modeled by the system (stream, elastic and/or adaptive), and number of traffic classes per traffic type (e.g., one or more elastic classes). The second one is the computational complexity of the numerical method, and the third one is the accuracy of the given numerical method. For the model that is considered in the present chapter, there exists a numerical method that gives exact results, see Rácz et al. (2001). However, that method has very high computational complexity, because it solves the underlying Markov chain with $\mathrm{O}(C^K)$ states, where C is the link capacity and K is the number of traffic classes. If we do not insist on having multiple classes per traffic type, there are less complex methods, see Delcoigne et al. (2004), Litjens et al. (2008). If we do not consider both stream and elastic traffic together, but either stream or elastic traffic classes separately, there are also less complex methods available, see Dziong & Roberts (1987), Kaufman (1981), and Koukoulidis (1993). In the present chapter, we study the multi-class stream-elastic system and we propose a method with significantly lower computational complexity (solving a Markov chain with $\mathrm{O}(C^2)$ states) than the exact solution by Rácz et al. (2001). On the other hand the model is detailed enough to contain multiple classes and the integration of stream and elastic traffic types. With the accuracy demonstrated in Section 5.4, the proposed method provides better tradeoffs between computational complexity, accuracy and level of model details than

alternative solutions.

This chapter is organized as follows. Section 5.1 describes the system and introduces the considered performance measures. Section 5.2 introduces the two-dimensional representation of the model. Section 5.3 provides the method that calculates the considered performance measures. In Section 5.4 we demonstrate the accuracy of the method and Section 5.5 concludes this chapter.

5.1 System description and model assumptions

This section describes the behavior of the modelled system and characterizes the traffic demands that arrive to the system. Let us first focus on the characteristics of the traffic flows. They belong to one of the following two traffic flow categories:

- Class-i *stream* flows are characterized by their peak rate b_i, flow arrival rate λ_i and flow departure rate μ_i.

- Class-i *elastic* flows are characterized by their peak rate limitation b_i, flow arrival rate λ_i, and ideal departure rate μ_i. The ideal departure rate is experienced when the peak rate is available throughout the whole time the flow receives service. The actual instantaneous departure rate is proportional to the service rate received by the flows.

An intuitive validation of the peak rate limitation can be for example the existence of a low bandwidth access link leading to the resource under study.

In order to ensure quality of service when there is not enough free capacity, a resource sharing policy is needed. To accomplish this the system under study works according to the following rules:

- The stream flows always get their peak rates. The elastic flows are only allowed to use the capacity left by the stream flows.

- The sum of the peak rates of ongoing stream flows has to be less then T_{st} and the sum of the peak rates of concurrent elastic flows has to be less then T_{el}. This means that arriving flows might be blocked. Note that if we want to analyze such a system where elastic flows are never blocked, T_{el} should be set to an adequately high value. In addition, the admission control on the stream flows (T_{st}) must be set in a way that there is always some service rate available for the elastic flows, i.e., $C > T_{st}$. This inequality is needed to keep the approximation as accurate as illustrated by Section 5.4. Both T_{st} and T_{el} are input parameters of the system.

- If the sum of the peak rates of all ongoing elastic flows does not exceed the capacity left by the ongoing stream flows, each flow gets its required peak rate.
- Otherwise, when the link is serving at its total capacity, the bandwidth left unoccupied by stream flows is shared between individual elastic flows in proportion to their peak rate, that is two flows from the same class always get the same amount of bandwidth, and a flow with two times higher peak rate limitation gets two times higher bandwidth as well.

Note that this system is not equivalent with the peak-rate limited Discriminatory Processor Sharing (DPS) model considered in Chapter 4. This system is a special case of DPS, where weights are equal to the peak rates.

System descriptors		
C	link capacity	$[BU]$
T_{st}	limit on total stream occupancy	$[BU]$
T_{el}	limit on total elastic occupancy assuming peak rates	$[BU]$

Descriptors of the i-th traffic class		
TYPE	stream or elastic class	-
b_i	peak rate (limitation)	$[BU]$
λ_i	flow arrival rate	$[flows/sec]$
t_i	mean flow holding time; $\mu_i = 1/t_i$	$[sec]$
ρ_i	$\rho_i = \lambda_i/\mu_i$ (and $\rho_i b_i$ is the offered load)	

Table 5.1: Input parameters

This model is Markovian under the assumptions that all arrival processes form independent Poisson processes and both the service times of stream flows and the offered load of the elastic flows are exponentially distributed (Fodor et al. (2002)). Let us call the states of this Markovian model as micro-states. The number of micro-states grows exponentially with the number of traffic classes.

Table 5.1 summarizes the input parameters describing the system. Capacity related parameters, such as peak rates and the link capacity, are integers and have to be set in bandwidth units (BU). Note that the integer system description supports rational numbers in general by selecting the greatest common divisor as BU. Table 5.2 summarizes the notations used.

5.1.1 Performance measures

The most important performance measure of elastic classes is the throughput. The statistical properties of throughput are different for each elastic class in a system with multiple elastic classes. Therefore, there is a need to define throughput measures on a per elastic class basis. Let us define the term average throughput (θ_i) of class i flows as the average class i flow size over the average holding times of class i flows in steady-state of the system. This also means that the perceived throughputs of individual flows are weighted by the amount of bytes the flows have to transmit. This throughput measure is aligned to the observation that longer flows play a more significant role in the performance experienced by users.

$$\theta_i = \frac{b_i \frac{1}{\mu_i}}{E(t_i)} = \frac{\overbrace{\lambda_i\ (1-B_i)\ b_i \frac{1}{\mu_i}}^{Avg\ incoming\ rate}}{\underbrace{\lambda_i\ (1-B_i)\ E(t_i)}_{Avg\ number\ of\ flows\ (Little's\ law)}} = \frac{\overbrace{b_i \sum_{\forall \underline{n} \in \Omega} \Pi(\underline{n})\ n_i\ r(\underline{n})}^{=Avg\ service\ rate}}{\sum_{\forall \underline{n} \in \Omega} \Pi(\underline{n})\ n_i} \tag{5.1}$$

As equation (5.1) shows the average instantaneous throughput of class i flows is equivalent of the average throughput measure defined above. As a consequence, the average throughputs can be calculated based on the stationary distribution of the underlying Markovian process. The formulation of the throughput calculation is derived in Section 5.3.4.

5.2 Macro-state model

In order to calculate the average throughputs, we introduce the macro-state model. The macro-state model provides a means to describe the original multi-rate stream-elastic system with fewer details based on aggregated statistical descriptors.

Macro-state (k, j) represents all *micro-states* where the service rate allocated to stream flows is k and the sum of peak rate limits of elastic flows adds up to j. In a given macro-state the bandwidth left by stream flows is constant, i.e. $C - k$ and the bandwidth reduction of elastic flows stays also the same, i.e. $\min(1, \frac{C-k}{j})$. Figure 5.1 illustrates the macro-state model.

Balance equations for macro-states (5.2) can be derived from the global balance equations of the corresponding micro-states, that is the global balance equations need to be summed. See Table 5.1 and Table 5.2 for notations. Equation (5.2) compresses the original linear global balance equations into significantly fewer, but non-linear equations.

Notations for models	
K	number of traffic classes
K_{st}	set of the stream traffic classes
K_{el}	set of the elastic traffic classes
$\underline{b}$	peak rates; $[b_1, b_2, \ldots, b_K]$
$\underline{n} = \{n_1, n_2, \ldots, n_K\}$	micro-state; n_i is the number of flows from class-i
Ω	state space of micro-states; $\left\{\underline{n} \in \mathbb{Z}^K : \sum_{i\in K_{st}} n_i b_i \leq T_{st},\ \sum_{i\in K_{el}} n_i b_i \leq T_{el}\right\}$
$\Pi(\underline{n})$	stationary probability of a micro-state
$\mathcal{S}$	state space of macro-states; $\{\{\alpha_{st}, \alpha_{el}\} \in \mathbb{Z}^2 :\ \alpha_{st} \leq T_{st}, \alpha_{el} \leq T_{el}\}$
$P(k,j)$	stationary probability of a macro-state; (joint resource occupancy distribution) $P(k,j) = \Pr(\mathcal{B}_{st} = k, \mathcal{B}_{el} = j)$
$r(\underline{n})$	bandwidth reduction of elastic flows in a micro-state; $\min\left(1, \frac{C - \sum_{i\in K_{st}} n_i b_i}{\sum_{i\in K_{el}} n_i b_i}\right)$
$r(k,j)$	bandwidth reduction of elastic flows in a macro-state; $\min\left(1, \frac{C-k}{j}\right)$
$E(n_i\|k,j)$	mean number of flows of class-i in a macro-state; $E(n_i\|k,j) = \mathbb{E}(\mathcal{N}_i\|\ \mathcal{B}_{st} = k, \mathcal{B}_{el} = j)$
B_i	blocking probability of class-i flows;

Table 5.2: Notations used

$$P(k,j)\Bigg(\underbrace{\sum_{i\in K_{el}\cup K_{st}} \lambda_i}_{Flow\ arrival} + \underbrace{\sum_{i\in K_{st}} E(n_i|k,j)\mu_i}_{Departure\ of\ stream\ flow} + \underbrace{r(k,j)\sum_{i\in K_{el}} E(n_i|k,j)\mu_i}_{Departure\ of\ elastic\ flow}\Bigg) =$$

$$\underbrace{\sum_{i\in K_{st}} P(k-b_i,j)\lambda_i + \sum_{i\in K_{el}} P(k,j-b_i)\lambda_i}_{Flow\ arrival} + \underbrace{\sum_{i\in K_{st}} P(k+b_i,j)E(n_i|k+b_i,j)\mu_i}_{Departure\ of\ stream\ flow} +$$

$$\underbrace{\sum_{i\in K_{el}} P(k,j+b_i)\ r(k,j+b_i)\ E(n_i|k,j+b_i)\mu_i}_{Departure\ of\ elastic\ flow} \quad (5.2)$$

Note that, if the conditional expectations are given, equation (5.2) falls back to a

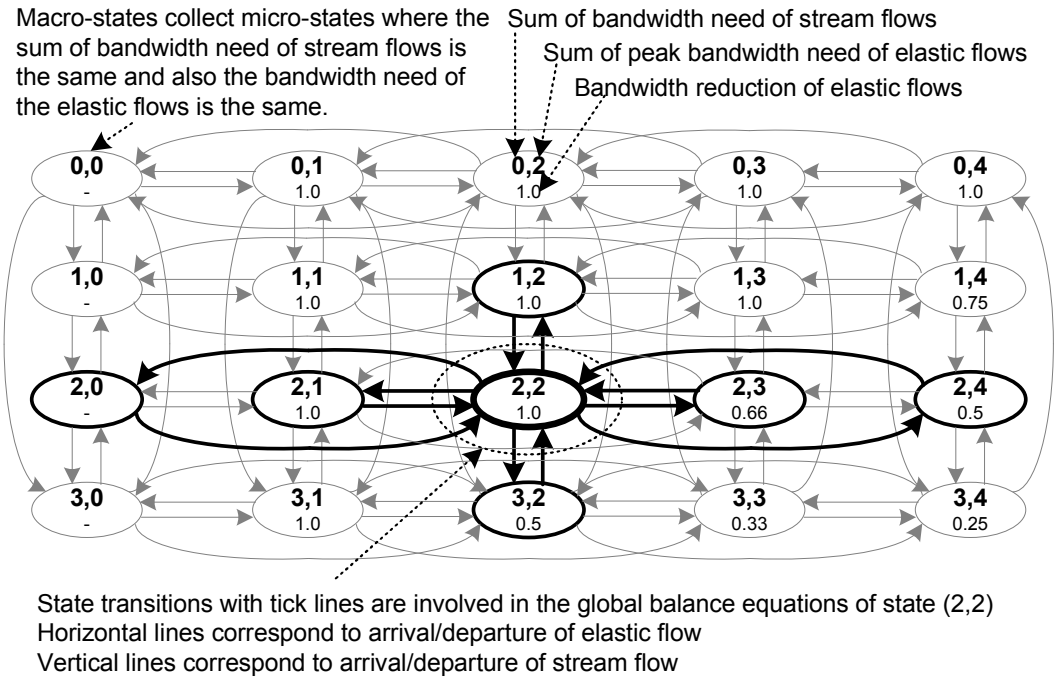

Figure 5.1: Illustration of macro-states model. $C = 4$, $T_{st} = 3$, $T_{el} = 4$.

set of linear equations that specifies the stationary probabilities of the macro-states.

5.3 Method

In this section we propose a method to approximate the average throughput of each elastic class. The method is based on the macro-state model and is derived from characteristics of the underlying stream-elastic system. The method is comprised of the following steps:

1. The conditional expectations of the number of stream flows from a given stream class in a given macro-state are approximated for all stream classes and for all macro-states.

2. The conditional expectations of the number of elastic flows from a given elastic class in a given macro-state are approximated for all elastic classes and for all macro-states.

3. Stationary probabilities of the macro-states are calculated by solving a set of linear equations. The conditional expectations of the number of flows in different classes are used as additional input.

4. The average throughput for each elastic class is derived from descriptors of the macro-state model.

The macro-state based method reduces computational complexity by reducing the state space. The number of states in the micro-state model grows exponentially with the number of traffic classes on the one hand $O(c^{(K_{st}+K_{el})})$, whereas the number of states in the macro-state model does not depend on the number of traffic classes on the other hand $(T_{st}+1)\times(T_{el}+1)$. Therefore the computational complexity of the proposed method is equivalent of solving $(T_{st}+1)\times(T_{el}+1)$ linear equations. The following subsections detail each step of the method individually.

5.3.1 Conditional expectations for stream flows

Conditional expectations of the number of stream flows are approximated by modeling the stream classes independently of elastic flows, i.e., a stream only model is built. This allows us to calculate stream flow characteristics in a recursive way. The approximation is formalized by equations (5.3), (5.4) and (5.5).

The conditional expectation of the number of class i stream flows ($i \in K_{st}$) in macro-state (k,j) can be approximated by:

$$E(n_i|k,j) \approx E(\hat{n}_i|k) \tag{5.3}$$

where $E(\hat{n}_i|k)$ satisfies:

$$E(\hat{n}_i|k)\hat{P}(k) = \rho_i \hat{P}(k-b_i) \tag{5.4}$$

and $\hat{P}(k)$ is calculated by:

$$k\hat{P}(j) = \sum_{i\in K_{st}} \rho_i b_i \hat{P}(k-b_i) \tag{5.5}$$

with the initial condition that $\hat{P}(k) = 0$ for all $k < 0$.

The observation that the conditional expectations of steam classes are approximated accurately by calculating them based on a stream only system (5.3) is a characteristic of the stream-elastic system. The accuracy of this observation is supported numerically in Section 5.4, which shows that the average throughput approximations are accurate in a given range. Stream flows by themselves form a Markov chain as stream flow departure rates do not depend on elastic flows. The stream only Markov chain is reversible and the well known Kaufman-Roberts formula (5.5) calculates the stationary probabilities of the corresponding macro-states. The equation of conditional expectations for stream flows (5.4) also follows from local balance equations of the stream only Markov chain.

5.3.2 Conditional expectations for elastic flows

Conditional expectations of elastic flows are approximated by assuming that the bandwidth taken by stream flows is constant throughout the lifetime of elastic flows. This allows us to approximate the conditional expectations for elastic classes recursively. At the same time the dependency between elastic and stream flows is kept. The approximation is formalized by equations (5.6), (5.7) and (5.8).

The conditional expectation number of class i elastic flows ($i \in K_{el}$) in macro-state (k, j) can be approximated by:

$$E(n_i|k,j) \approx E(\tilde{n}_i|k,j) \tag{5.6}$$

where $E(\tilde{n}_i|j,k)$ satisfies:

$$min\left(\frac{C-k}{j}, 1\right) E(\tilde{n}_i|k,j)\tilde{P}(k,j) = \rho_i \tilde{P}(k, j-b_i) \tag{5.7}$$

and $\tilde{P}(k,j)$ is calculated by:

$$min(C-k,j)\tilde{P}(k,j) = \sum_{i \in K_{el}} \rho_i b_i \tilde{P}(k, j-b_i) \tag{5.8}$$

with the initial condition that $\tilde{P}(k,j) = 0$ for all $j < 0$ and for any k.

The approximation in (5.6) corresponds to the special case when the departure rates of all elastic classes are at least a magnitude higher than the departure rates of any of the stream classes. The observation that the approximation is accurate in a more general case is a characteristic of the stream-elastic system and it is supported by the numerical analysis of Section 5.4, which shows that the average throughput approximations are accurate in a given range. Equations (5.7) and (5.8) model an elastic class only system with constant capacity of $C - k$, which is the capacity available to elastic flows in the corresponding row of macro-states. Equations (5.7) and (5.8) were developed by Koukoulidis (1993) to approximate characteristics of multi-class elastic systems.

5.3.3 Stationary probabilities of macro-states

Stationary probabilities are calculated by solving the set of linear equations defined in (5.2) in Section 5.2. After approximating $E(n_i|k,j)$ with method proposed in Section 5.3.1 and Section 5.3.2, the linear equation system defined by (5.2) for each macro-state enables us to compute the macro-states' stationary probability distribution. This could be done by using one of the standard methods (see Stewart (1994)) to solve the $(T_{el}+1) \times (T_{st}+1)$ linear equations.

Let us present a more efficient solution method which exploits the special structure of the equations describing the connection between the stationary probabilities of the macro-states. The most time consuming step of the method is the solution of $b_m \times (T_{st}+1)$ linear equations, where m is the index of the elastic class which has the largest peak rate. The following theorem defines the main steps of the proposed solution method. Note that this method needs accurate mathematical tools.

Theorem 5.3.1. *The stationary probability of the macro-states can be efficiently evaluated using the following three steps:*

- *expressing the probability of all macro-states as a linear combination of the probability of macro-states in* $\mathcal{S}_0 = \{\{k,j\} : \ j = 0 \dots b_m - 1, \ k = 0 \dots T_{st}\}$*; in this step we use the global-balance equations of macro-states* $\{\{k,j\} : \ j = 0 \dots T_{el} - b_m, \ k = 0 \dots T_{st}\}$*;*
- *determining the probabilities of states in* $\mathcal{S}_0$*; in this step we use the global-balance equations of states* $\{\{k,j\} : \ j = T_{el} - b_m + 1 \dots T_{el}, \ k = 0 \dots T_{st}\}$ *and solve a linear system of equations of size* $b_m \times (T_{st}+1)$*.*
- *determining the probabilities of states in* $\mathcal{S} \setminus \mathcal{S}_0$ *using the probabilities of states in* $\mathcal{S}_0$*.*

Proof of Theorem 5.3.1: According to (5.2), the global balance equation of the macro-state $\{k, j-b_m\}$ can be rewritten as:

$$
\begin{aligned}
&P(k,j-b_m)\left[\sum_{i\in K_{st}}(\lambda_i + E(n_i|k,j-b_m)\mu_i) + \sum_{i\in K_{el}}(\lambda_i + r(k,j-b_m)E(n_i|k,j-b_m)\mu_i)\right] = \\
&\sum_{i\in K_{st}}[P(k-b_i,j-b_m)\lambda_i + P(k+b_i,j-b_m)E(n_i|k+b_i,j-b_m)\mu_i] + \\
&\sum_{i\in K_{el}}[P(k,j-b_i-b_m)\lambda_i + P(k,j+b_i-b_m)r(k,j+b_i-b_m)E(n_i|k,j+b_i-b_m)\mu_i]
\end{aligned}
\tag{5.9}
$$

The next step is to derive $P(k,j),\ j \geq b_m$ from the global-balance equation of macro-state $\{k, j-b_m\}$:

$$
\begin{aligned}
P(k,j) = &\frac{1}{E(n_m|k,j)\mu_m} \times \\
&\left(P(k,j-b_m)\left[\sum_{i\in K_{st}}(\lambda_i + E(n_i|k,j-b_m)\mu_i) + \sum_{i\in K_{el}}(\lambda_i + r(k,j-b_m)E(n_i|k,j-b_m)\mu_i)\right] - \right. \\
&\sum_{i\in K_{st}}[P(k-b_i,j-b_m)\lambda_i + P(k+b_i,j-b_m)E(n_i|k+b_i,j-b_m)\mu_i] - \\
&\left.\sum_{i\in K_{el}}P(k,j-b_i-b_m)\lambda_i - \sum_{i\in K_{el}, i\neq m}P(k,j+b_i-b_m)r(k,j+b_i-b_m)E(n_i|k,j+b_i-b_m)\mu_i\right)
\end{aligned}
\tag{5.10}
$$

The values of $P(k,j)$'s for a fixed j can be expressed as a linear combination of $P(k,i)$'s for $j-2b_m \leq i < j$. The structure of state transitions allows this representation. Introducing the notation $\underline{P}(j) = [P(k,j)]_k$ (as a row vector, where $k = 0, 1, \ldots, T_{st}$) this linear combination has the form of:

$$\underline{P}(j) = \sum_{i=j-2b_m}^{j-1} \underline{P}(i) \cdot \mathbf{G}^{(i)}(j). \tag{5.11}$$

where the matrix $\mathbf{G}^{(i)}(j)$ shows how $P(k,j)$s depend on $P(k,i)$s. We introduce the matrix $\mathbf{G}(j)$ which has the following block structure:

$$\mathbf{G}(j) = \begin{pmatrix} \mathbf{G}^{(0)}(j) \\ \mathbf{G}^{(1)}(j) \\ \vdots \\ \mathbf{G}^{(b_m-1)}(j) \end{pmatrix},$$

where $\mathbf{G}^{(i)}(j)$ is a square matrix of size $T_{st}+1$. Using $\mathbf{G}(j)$, the $\underline{P}(j)$ can be expressed using $\underline{P}(0),\ \underline{P}(1), \ldots, \underline{P}(b_m-1)$ as follows:

$$\underline{P}(j) = \sum_{i=j-2b_m}^{j-1} \underline{P}(i) \cdot \mathbf{G}^{(i)}(j) = \underline{P}_0 \cdot \mathbf{G}(j), \tag{5.12}$$

where $\underline{P}_0$ is a row vector that consists of $\underline{P}(0),\ \underline{P}(1), \ldots, \underline{P}(b_m-1)$. By definition of $\mathbf{G}(j)$:

$$\mathbf{G}(0) = \begin{pmatrix} \mathbf{I} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix}, \quad \mathbf{G}(1) = \begin{pmatrix} \mathbf{0} \\ \mathbf{I} \\ \vdots \\ \mathbf{0} \end{pmatrix}, \quad \ldots, \ \mathbf{G}(b_m-1) = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{I} \end{pmatrix}.$$

We can rewrite the global balance equations (5.2) for macro-states $(k, j-b_m)$, where

$k = 0, \ldots, T_{st}$ using equation (5.12) in matrix form as follows:

$$\underbrace{\underline{P}_0 \cdot \mathbf{G}(j - b_m) \cdot \mathbf{A}(j - b_m)}_{Outgoing\ from\ macro-states\ (k,j-b_m)} = \underbrace{\underline{P}_0 \cdot \mathbf{G}(j - b_m) \cdot \mathbf{B}(j - b_m)}_{Incoming\ from\ macro-states\ (k,j-b_m)} + \tag{5.13}$$
$$+ \underbrace{\sum_{i \in K_{el}} \underline{P}_0 \cdot \mathbf{G}(j - b_m - b_i) \cdot \mathbf{C}(i)}_{Incoming\ from\ macro-states\ (k,j-b_m-b_i)} +$$
$$+ \underbrace{\sum_{i \in K_{el}} \underline{P}_0 \cdot \mathbf{G}(j - b_m + b_i) \cdot \mathbf{D}(j - b_m + b_i, i)}_{Incoming\ from\ macro-states\ (k,j-b_m+b_i)}.$$

From the equation above we can get $\mathbf{G}(j)$ for $j \geq b_m$:

$$\begin{aligned} \mathbf{G}(j) &= \\ &= \Big[\mathbf{G}(j - b_m) \cdot \mathbf{A}(j - b_m) - \\ &\quad \mathbf{G}(j - b_m) \cdot \mathbf{B}(j - b_m) - \\ &\quad \sum_{i \in K_{el}} \mathbf{G}(j - b_m - b_i) \cdot \mathbf{C}(i) - \\ &\quad \sum_{i \in K_{el}, i \neq m} \mathbf{G}(j - b_m + b_i) \cdot \mathbf{D}(j - b_m + b_i, i) \Big] \cdot \mathbf{D}^{-1}(j, i). \end{aligned} \tag{5.14}$$

The structure of the matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ are as follows, each of them is a square matrix of size $T_{st} + 1$. The $\mathbf{A}$ matrix is a diagonal matrix with the following structure:

$$\mathbf{A}(j) = \begin{pmatrix} \sum_{i \in K_{el}} \lambda_i & 0 & \ldots & 0 \\ 0 & \sum_{i \in K_{el}} \lambda_i & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & \sum_{i \in K_{el}} \lambda_i \end{pmatrix} +$$

$$\begin{pmatrix} r(0,j) \sum_{i \in K_{el}} E(n_i|0,j)\mu_i & 0 & \ldots & 0 \\ 0 & r(1,j) \sum_{i \in K_{el}} E(n_i|1,j)\mu_i & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & r(T_{st},j) \sum_{i \in K_{el}} E(n_i|T_{st},j)\mu_i \end{pmatrix}.$$

The **B** matrix is the sum of an upper triangular matrix which contains the arrival intensities ($\lambda_i,\ i \in K_{st}$) of stream classes and a lower triangular matrix which contains the departures intensities ($E(n_i|k,j)\mu_i,\ i \in K_{st}$) of stream classes. The position of elements depends on the peak rate of the corresponding stream class (b_i). For example, $b_{st_1} = 1, b_{st_2} = 3$:

$$\mathbf{B}(j) = \begin{pmatrix} 0 & \lambda_{st_1} & 0 & \lambda_{st_2} & 0 & \dots \\ 0 & 0 & \lambda_{st_1} & 0 & \lambda_{st_2} & \dots \\ \vdots & 0 & 0 & \ddots & \ddots & \dots \end{pmatrix} +$$

$$+ \begin{pmatrix} 0 & 0 & \dots & 0 \\ E(n_{st_1}|1,j)\mu_{st_1} & 0 & \dots & 0 \\ 0 & E(n_{st_1}|2,j)\mu_{st_1} & 0 & \dots \\ E(n_{st_2}|3,j)\mu_{st_2} & 0 & E(n_{st_1}|3,j)\mu_{st_1} & \dots \\ \vdots & \ddots & \ddots & \ddots \end{pmatrix}.$$

Matrices **C** and **D** are diagonal matrices:

$$\mathbf{C}(i) = \begin{pmatrix} \lambda_i & 0 & \dots & 0 \\ 0 & \lambda_i & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \lambda_i \end{pmatrix}.$$

$$\mathbf{D}(j,i) = \begin{pmatrix} r(0,j)E(n_i|0,j)\mu_i & 0 & \dots & 0 \\ 0 & r(1,j)E(n_i|1,j)\mu_i & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & r(T_{st},j)E(n_i|T_{st},j)\mu_i \end{pmatrix}.$$

Finally, we substitute the $\underline{P}(j),\ \ T_{el} - b_m < j \leq T_{el}$ into the remaining global-balance equations and the $\underline{P}(j) = \underline{P}_0 \cdot \mathbf{G}(j)$ identity results in a linear system of equations in probabilities of states $\mathcal{S}_0$. □

5.3.4 Throughput calculation

The average throughput of class i flows is calculated from the macro-state representation of the system. This is possible because the instantaneous throughputs of elastic flows depend only on the sum of the resource demands of elastic flows and the sum of the resource demands of the stream flows. Consequently, the instantaneous throughput in a macro-state is the same regardless of the micro-states. Equation (5.15) formalizes how the average class i throughputs can be calculated from the macro-states' stationary probabilities and the conditional expectations. Equation (5.15) is based on the fact that the average throughput (θ_i) is equivalent of the instantaneous throughput as we discussed in Section 5.1.1.

$$\theta_i = \frac{\sum\limits_{\{k,j\}\in\mathcal{S}} P(k,j)\ E(n_i|k,j)\ r(k,j)\ b_i}{\sum\limits_{\{k,j\}\in\mathcal{S}} P(k,j)\ E(n_i|k,j)} \tag{5.15}$$

5.4 Accuracy

In this section I demonstrate that the method proposed in Section 5.3 provides an accurate approximation where the holding time of elastic classes are the same and the holding time of stream classes are the same. In addition, we investigate how the error of the approximation depends on different system parameters.

The error comes from the fact that classes have different peak rates. If we have stream classes with the same peak rate and elastic classes with another peak rate, the method provides exact results.

5.4.1 Simulations

The exact values of the average throughputs (θ) and the blocking probabilities (B) were derived by simulations. We use the results of $\sim$ 10 000 random scenarios to demonstrate the accuracy of the method. The systems under investigation have four classes, class-1 and class-2 are stream classes and class-3 and class-4 are elastic classes. The system is scaled in terms of offered load so that we always get scenarios where the throughput of a flow is smaller than its peak rate, i.e., flows are compressed. The settings of the input parameters during the simulations can be found in Table 5.3. Values for each simulation scenarios were randomly selected from the ranges indicated. The holding time of both stream classes were the same($\mu_{st1} = \mu_{st2}$) and the holding time of both elastic classes were also the same ($\mu_{el1} = \mu_{el2}$).

5.4.2 Error Measures

To quantify the accuracy of the method we introduce the following error measures: the *per-class error* (e_i), the *average per-class error* (e_A) and the *system throughput error* (e_B). The per-class error of the throughput is defined as:

$$e_3 = \frac{\widetilde{\theta}_3 - \theta_3}{\theta_3} \qquad e_4 = \frac{\widetilde{\theta}_4 - \theta_4}{\theta_4}.$$

where $\widetilde{\theta}_3$ and $\widetilde{\theta}_4$ denotes the performance measures provided by the method for the two elastic classes. θ_3 and θ_4 denotes the exact values of average throughputs.

In addition, let us define the average per-class error as the average of per-class errors weighted by the carried load of elastic classes:

$$e_A = \sqrt{\frac{(1-B_3)\rho_3 b_3 e_3^2 + (1-B_4)\rho_4 b_4 e_4^2}{(1-B_3)\rho_3 b_3 + (1-B_4)\rho_4 b_4}}.$$

where B_3, B_4 denotes the blocking probabilities.

The system throughput error is defined as:

$$e_B = \frac{\widetilde{\theta}_{avg} - \theta_{avg}}{\theta_{avg}}.$$

where,

$$\theta_{avg} = \frac{(1-B_3)\rho_3 b_3 \theta_3 + (1-B_4)\rho_4 b_4 \theta_4}{(1-B_3)\rho_3 b_3 + (1-B_4)\rho_4 b_4} \qquad \widetilde{\theta}_{avg} = \frac{(1-B_3)\rho_3 b_3 \widetilde{\theta}_3 + (1-B_4)\rho_4 b_4 \widetilde{\theta}_4}{(1-B_3)\rho_3 b_3 + (1-B_4)\rho_4 b_4}$$

5.4.3 Simulation Results

As illustrated in Figure 5.2-Figure 5.5, the method provides a good approximation if the system parameters from Table 5.3 are used. Both the values of the average per-class error and the system throughput error are concentrated at zero in Figure 5.2 and Figure 5.3, respectively. Moreover, the system throughput error is less than the average per-class error.

Figure 5.4, Table 5.5 and Table 5.6 show that the method is more accurate for larger systems. To verify this, we grouped the random scenarios by the level of compression (i.e. bandwidth reduction) of elastic classes. The average compression is defined as follows:

$$\bar{r} = \frac{(1-B_3)\rho_3 \frac{\theta_3}{b_3} + (1-B_4)\rho_4 \frac{\theta_4}{b_4}}{(1-B_3)\rho_3 + (1-B_4)\rho_4} \tag{5.16}$$

$\bar{r} = 1$ means no compression. Based on the value of $\bar{r}$ we defined three compression

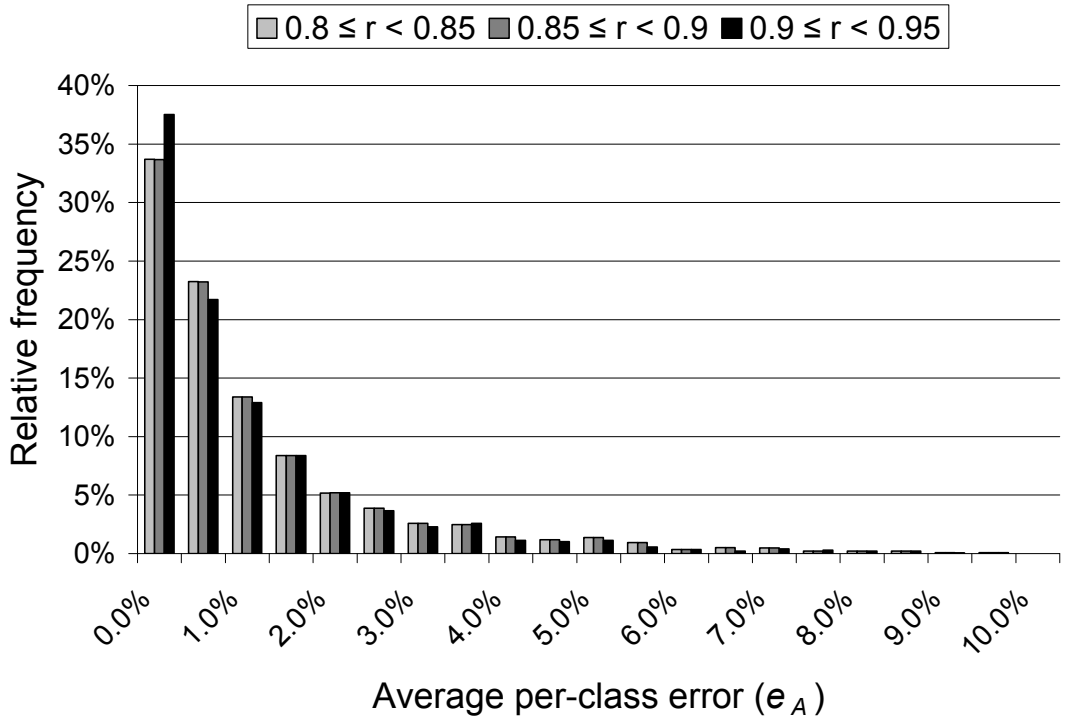

Figure 5.2: Histogram of the average per-class error, e_A

groups: $0.8 \leq \bar{r} < 0.85$, $0.85 \leq \bar{r} < 0.9$, and $0.9 \leq \bar{r} < 0.95$. Figure 5.4 shows the average per-class error for these three groups. *The error decreases for each group if we increase the capacity.*

Figure 5.5, Table 5.5 and Table 5.6 show that the method provides more accurate results for less compressed elastic flows. The figure depicts the per-class error of the second elastic class versus the per-class error of the first elastic class. Scenarios with the least compressed elastic flows have the smallest errors. Values in the figure are concentrated around zero. The higher the level of compression, the bigger the average distance from zero. If elastic flows are never compressed the method has no error, since in this case elastic flows can be considered as stream flows from a modeling point of view. For a stream-only system the proposed approximation method falls back to the Kaufman-Roberts formula (5.5), which provides exact results.

Table 5.4 shows how the average per-class error depends on input parameters. The strongest correlation can be noticed with the fraction of stream load in the total traffic ($frac_{st}$). The higher this fraction is the bigger the error of the proposed method gets. Capacity shows the second strongest correlation in absolute value. Negative correlation means in this case that as link capacity increases, the average per-class error decreases as it is shown in Figure 5.4. The offered load is in the second strongest positive correlation with the error. Correlation with the peak rate of both elastic classes (Peak rate-3 and Peak rate-4) is also significant. Note that correlation with Peak rate-3 and Peak rate-4 is not the same because of the relatively small

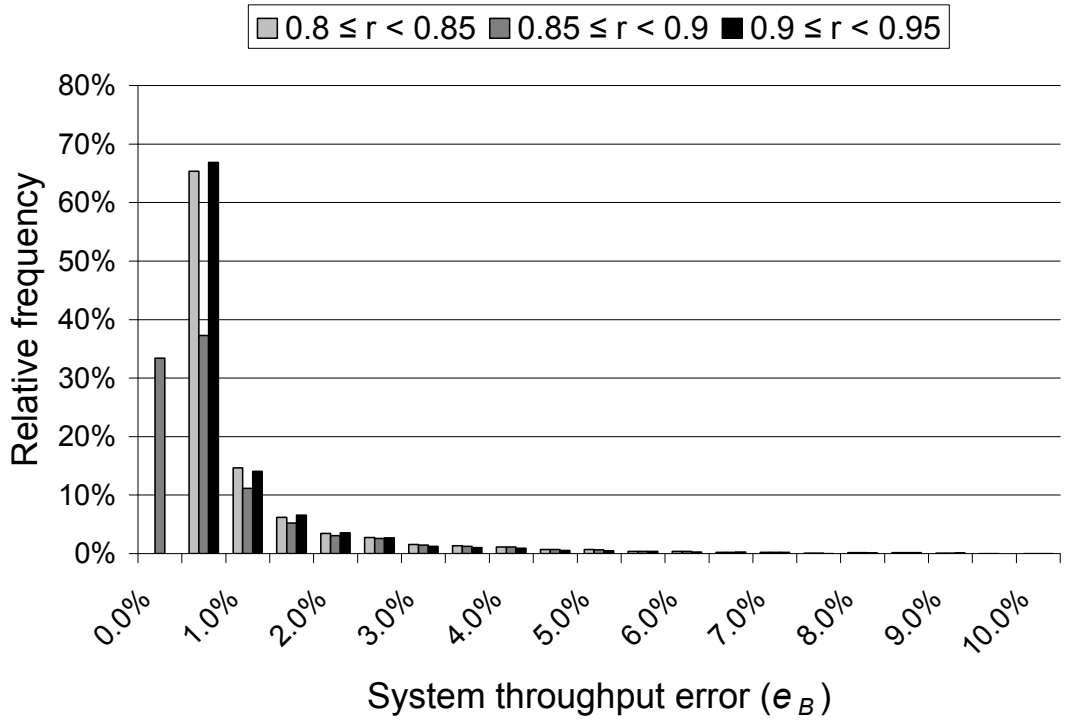

Figure 5.3: Histogram of the system throughput error, e_B

number of random scenarios. The ratio of the holding times of the elastic classes and the stream classes (μ_{st}/μ_{el}) has still some impact on the average per-class error, but not as high as in the case of the three aforementioned input parameters.

5.4.3.1 Average error values and correlation

This section presents additional accuracy evaluation results of the proposed method.

Table 5.5 shows the error measures for all scenarios and Table 5.6 shows the error measures for scenarios where the link capacity is higher than 300 BU.

Table 5.7 shows how the average per-class error depends on input parameters. In this table only scenarios are considered where the link capacity is higher than 300 BU.

5.5 Conclusion

We have proposed a method to approximate the average throughput of elastic classes efficiently in a multi-rate system. We proposed a grouping of micro-states that provides a means to decompose the throughput calculation into stand alone steps. Consequently, the proposed method is less complex. We have shown that by applying known recursive formulae, the conditional expectations can be approximated. The proposed grouping also ensures that with the determined conditional expectations as

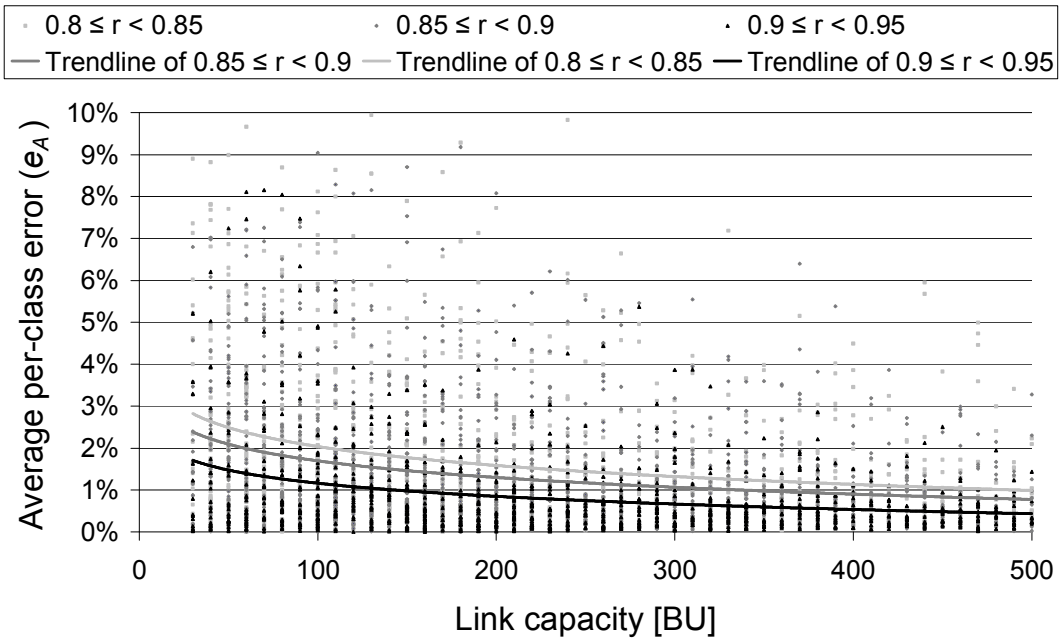

Figure 5.4: Average per-class error (e_A) vs. capacity

input, the macro-state probabilities can be calculated by solving a system of linear equations.

We have demonstrated that the method is accurate to be used in practice. Furthermore, applications can take advantage from the fact that the accuracy is improving as the capacity grows. The proposed method provides better tradeoffs between computational complexity, accuracy and level of model details than alternative solutions.

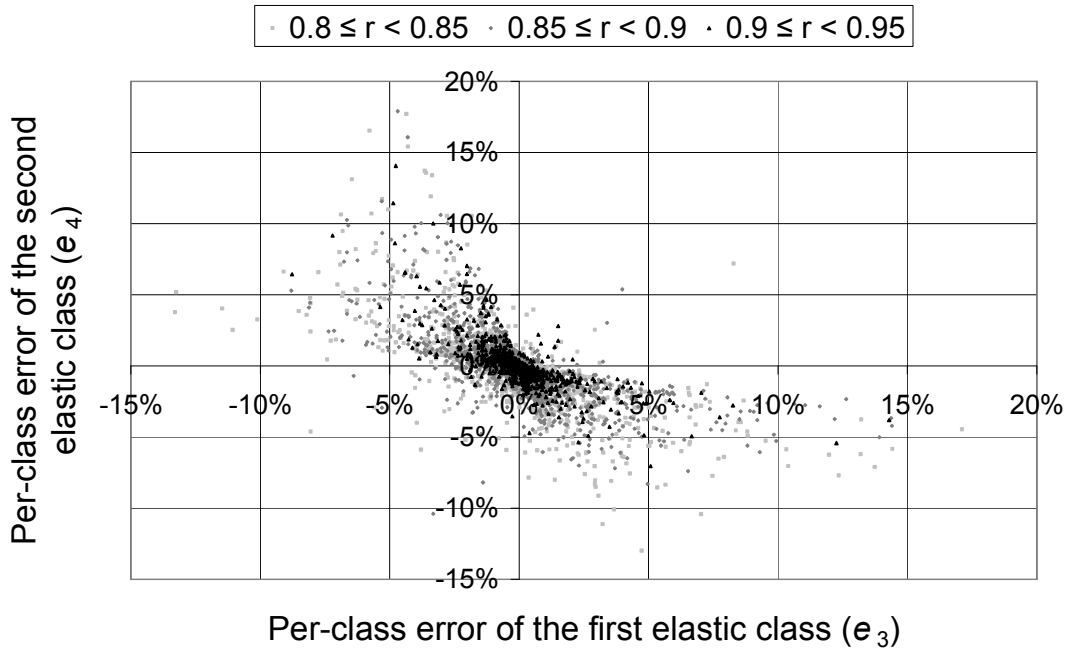

Figure 5.5: Per-class error of the second vs. the first elastic class, (e_3, e_4)

System descriptors			
C	link capacity	30, 31, 32,. . . , 500	$[BU]$
T_{st}	limit on total stream occupancy	$\min(0.9, (frac_{st} + 0.4)C$	$[BU]$
T_{el}	limit on total elastic occupancy assuming peak rates	$4C$	$[BU]$

Descriptors of the i-th traffic class			
TYPE	stream or elastic class		-
b_i	peak rates	1, 2, 3,. . . , 20	$[BU]$
μ_{el}/μ_{st}	ratio of mean elastic and stream flow departure rate	$\frac{1}{1000}, \frac{1}{500}, \frac{1}{200}, \frac{1}{100}, \ldots, 100, 200, 500, 1000$	$[sec]$
$\rho_i b_i$	offered load	0.2, 0.3, 0.4, 0.5, 0.6, 0.65,. . . 0.95, 0.96, 0.97, 0.98	
$frac_{st}$	The fraction of stream load in the total traffic	0.2, 0.25, 0.3, 0.35,. . . , 0.8	-
$frac_{st1}$	fraction of load of first stream class in total stream load	0.2, 0.25, 0.3, 0.35,. . . , 0.8	-
$frac_{el1}$	Fraction of load of first elastic class in total elastic load	0.2, 0.25, 0.3, 0.35,. . . , 0.8	-

Table 5.3: System parameters used for the simulations

Input parameter	all	$0.9 \leq \bar{r} < 0.95$	$0.85 \leq \bar{r} < 0.9$	$0.8 \leq \bar{r} < 0.85$
$frac_{st}$	26.97%	27.57%	28.46%	27.92%
Capacity	-25.4%	-25.9%	-25.37%	-23.45%
Offered load	24.23%	24.21%	22.46%	19.0%
Peak rate-3	-21.52%	-24.35%	-22.46%	-22.15%
Peak rate-4	-21.86%	-21.63%	-23.92%	-22.96%
μ_{el}/μ_{st}	18.34%	24.5%	18.49%	17.1%

Table 5.4: Correlation of e_A with the main input parameters grouped by compression level, all scenarios

Error measure	all	$0.9 \leq \bar{r} < 0.95$	$0.85 \leq \bar{r} < 0.9$	$0.8 \leq \bar{r} < 0.85$
e_A	1.37%	0.87%	1.38%	1.71%
e_B	0.74%	0.50%	0.77%	0.87%
e_3	-0.06%	-0.01%	0.02%	-0.19%
e_4	-0.12%	0.01%	-0.13%	-0.21%

Table 5.5: Average error values grouped by the compression level, all scenarios

Error measure	all	$0.9 \leq \bar{r} < 0.95$	$0.85 \leq \bar{r} < 0.9$	$0.8 \leq \bar{r} < 0.85$
e_A	0.92%	0.59%	0.93%	1.22%
e_B	0.44%	0.28%	0.44%	0.59%
e_3	-0.17%	-0.10%	-0.10%	-0.34%
e_4	-0.32%	-0.13%	-0.38%	-0.41%

Table 5.6: Average error values grouped by the compression level, $C > 300$ BU

Input parameter	all	$0.9 \leq \bar{r} < 0.95$	$0.85 \leq \bar{r} < 0.9$	$0.8 \leq \bar{r} < 0.85$
$frac_{st}$	32.19%	30.75%	33.89%	38.86%
Capacity	-7.21%	-9.91%	-12.07%	-1.58%
Offered load	39.82%	33.77%	39.57%	28.03%
Peak rate-3	-19.2%	-24.2%	-18.57%	-22.51%
Peak rate-4	-25.78%	-28.83%	-30.62%	-23.68%
μ_{el}/μ_{st}	12.49%	16.46%	23.71%	9.73%

Table 5.7: Correlation of e_A with the main input parameters grouped by compression level, $C > 300$ BU

Chapter 6

Conclusion

The increasing number of mobile broadband users and their even more increasing data demands come with challenges that motivated the work presented in this dissertation. The elaboration of new congestion avoidance methods are needed in these mobile telecommunications networks to ensure the appropriate QoS in the rapidly saturating network environment. Since the radio part of the network is often enhanced faster than the terrestrial part due to cost-efficiency reasons, the terrestrial part may be a potential bottleneck for the whole network. That is why the present dissertation focuses on the RAN TN of HSPA and addresses fairness issues, presents efficient congestion control methods and solves open issues in dimensioning methods, which can be applied in this part of the network.

I have proposed a method which provides fairness-optimal initial shaping rate for incoming HSDPA flows based on ongoing HSDPA flows. I have shown that the proposed method improves the transient fairness characteristics at user arrivals and also the long-term average fairness. The method can be applied in a rate-based or window-based congestion control where flows share the same bottleneck. I have also given a general solution for fairness-optimal initial rate in the case of second-order fairness measures.

I have proposed a novel, non-standardized, cross-layer window-based HSDPA Transport Network Congestion Control. This solution is based on extending the RLC protocol with congestion control functionality and uses the standardized congestion detection and signalling framework along with the standardized shaper. I have shown that the described RLC-based method can be a possible solution to provide efficient HSDPA congestion control.

I have characterized the discriminatory processor sharing model with peak-rate limitations and bandwidth-efficient rate sharing. I have shown that at a given state space the peak-rate limited DPS system includes a reduced-capacity DPS system over the compressed classes. Based on this I have given a method to determine the set

of compressed and uncompressed classes of flows and their bandwidth shares. I have also shown that there is a strict order of classes in which they become compressed, and this order coincides with the order of ratios of the corresponding class weights and class peak rates.

Finally, I have performed a flow level analysis of an other bandwidth sharing model, the integration of stream and elastic traffic, which is an adequate model for voice and data services. The average throughput has been calculated for elastic traffic classes and a two-dimensional macro-state representation has been introduced for the micro-state model. I have characterized the stationary probabilities of macro states and evaluated the accuracy of the approximation method. I have demonstrated that the method is accurate to be used in practice. Furthermore, applications can take advantage from the fact that the accuracy is improving as the capacity grows.

Bibliography

3GPP (2001). TS 25.853 V4.0.0 (Delay budget within the access stratum).

3GPP (2004). TS 25.308 version 6.3.0 (UTRA High Speed Downlink Packet Access (HSDPA) Overall Description).

3GPP (2005a). TR 25.902 version 6.0.0 (UMTS Iub/Iur congestion control).

3GPP (2005b). TS 25.943 version 6.0.0 (Deployment aspects).

3GPP (2008a). TS 25.321 version 8.1.0 (Medium Access Control (MAC) protocol specification).

3GPP (2008b). TS 25.435 version 6.5.0 (UTRAN Iub interface user plane protocols for Common Transport Channel data streams).

3GPP (2008c). TS 25.435 version 6.5.0 (UTRAN Iub interface user plane protocols for Common Transport Channel data streams).

3GPP (2008d). TS 36 series, release 8 LTE (Evolved UTRA) and LTE-Advanced radio technolgy.

3GPP (2009). TS 25.322 V8.5.0 (Radio Link Control (RLC) protocol specification).

3GPP (2011). TS 25.306 version 11.0.0 (UE Radio Access capabilities).

Aczél, J. & Dhombres, J. (1989). *Functional Equations in Several Variables*. Cambridge Univ. Press, Cambridge.

Assaad, M., Jouaber, B., & Zeghlache, D. (2004). TCP performance over UMTS-HSDPA system. *Telecommunication Systems*, *27*(2-4), 371–391.

Avrachenkov, K., Ayesta, U., Brown, P., & Núnez-Queija, R. (2005). Discriminatory processor sharing revisited. In *IEEE Infocom 2005*, (pp. 784–795)., Miami, FL.

Ayesta, U. & Mandjes, M. (2009). Bandwidth-sharing networks under a diffusion scaling. *Annals Operation Research*, *170*(1), 41–58.

Bajzik, L., Korossy, L., Veijalainen, K., & Vulkan, C. (2006). Cross-layer back-pressure to improve HSDPA performance. In *Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on*, (pp. 1–5).

Bergman, J., Gerstenberger, D., Gunnarsson, F., & Strom, S. (2009). Continued HSPA evolution of mobile broadband. *Ericsson Review*.

Bíró, J., Bérczes, T., Körösi, A., Heszberger, Z., & Sztrik, J. (2013). Discriminatory processor sharing from optimization point of view. In *ASMTA*, (pp. 67–80).

Borst, S. & Hegde, N. (2007). Integration of streaming and elastic traffic in wireless networks. *Infocom*. Proc. of INFOCOM.

Burman, D. (1981). Insensitivity in queuing systems. *Adv. Appl. Prob. 13*, 846–859.

Cheung, S., van den Berg, H., & Boucherie, R. (2005). Decomposing the queue length distribution of processor-sharing models into queue lengths of permanent customer queues. *Performance Evaluation*, *62*(1-4), 100–116.

Chiu, D. & Jain, R. (1989). Analysis of the increase/decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN*, *17*(1), 1–14.

Chiu, D. M. (2000). Some observations on fairness of bandwidth sharing. In *Proc. of ISCC'00*, Antibes, France.

Cisco White Paper (2012). Cisco visual networking index: Global mobile data traffic forecast update, 2010–2015. `http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.pdf`. FLGD 10229.

Dahlman, E., Parkvall, S., Sköld, J., & Beming, P. (2008). *3G Evolution – HSPA and LTE for Mobile Broadband*. Academic Press.

Delcoigne, F., Proutiere, A., & Regnie, G. (2004). Modeling integration of streaming and data traffic. *Performance Evaluation*, *55*(3-4), 185–209.

Deng, J., Han, Y. S., & Liang, B. (2009). Fairness index based on variational distance. In *Proceedings of the 28th IEEE conference on Global telecommunications*, GLOBECOM'09, (pp. 3338–3343)., Piscataway, NJ, USA. IEEE Press.

Dziong, Z. & Roberts, J. W. (1987). Congestion probabilities in a circuit-switched integrated services network. *Performance Evaluation*, *7*(4).

Ekstrom, H., Furuskar, A., Karlsson, J., Meyer, M., Parkvall, S., Torsner, J., & Wahlqvist, M. (2006). Technical solutions for the 3G long-term evolution. *Communications Magazine, IEEE*, *44*, 38–45.

Ericsson (2011). Mini-link TN datasheet. EN/LZT 110 5216 R1 ETSI.

Ericsson (2012). Traffic and market data report. `http://www.ericsson.com/res/docs/2012/traffic_and_market_report_june_2012.pdf`.

Ericsson White Paper (2008). High-speed technologies for mobile backhaul. `http://www.ericsson.com/technology/whitepapers/pdf/High-speed-technology-mobile-backhaul.pdf`. 284 23-3119 Uen Rev B.

Ericsson White Paper (2011a). Differentiated mobile broadband. `http://www.ericsson.com/res/docs/whitepapers/differentiated_mobile_broadband.pdf`. 284 23-3147 Uen.

Ericsson White Paper (2011b). HSPA evolution – beyond 3GPP release 10. `http://www.ericsson.com/res/docs/whitepapers/WP-HSPA-Evolution.pdf`. 284 23-3156.

Fan, Z. (2006). Dimensioning bandwidth for elastic traffic. In *NETWORKING 2002: Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications*, (pp. 826–837).

Fayolle, G., Mitrani, I., & Iasnogorodski, R. (1980). Sharing a processor among many job classes. *J. ACM*, *27*(3), 519–532.

Fodor, G., Rácz, S., & Telek, M. (2002). On providing blocking probability- and throughput guarantees in a multi-service environment. *International Journal of Communication Systems*, *15*, 257–285.

Garriga, B., Dominguez, F., Serrano, C., Tenorio, S., & Asensio, E. (2009). QoS load differentiation application in a UTRAN live network. In *VTC Spring'09*, (pp. 1–8).

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. *SIGKDD Explor. Newsl.*, *11*(1), 10–18.

He, L. & Walrand, J. C. (2006). Pricing and revenue sharing strategies for internet service providers. *IEEE Journal on Selected Areas in Communications*, *24*(5), 942–951.

Heikkinen, M., Kivi, A., & Verkasalo, H. (2009). Measuring mobile peer-to-peer usage: Case finland 2007. In S. Moon, R. Teixeira, & S. Uhlig (Eds.), *Passive and Active Network Measurement*, volume 5448 of *Lecture Notes in Computer Science* (pp. 165–174). Springer Berlin / Heidelberg. 10.1007/978-3-642-00975-4_17.

Jain, R. (1995). Congestion control and traffic management in atm networks: Recent advances and a survey. *COMPUTER NETWORKS AND ISDN SYSTEMS*, 1723–1738.

Jain, R., Chiu, D., & Hawe, W. (1984). A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Technical Report TR-301, DEC Research.

Jiawei Han, M. K. (2006). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers.

Joseph, J. & Avital, R. (2010). Making mobile broadband accessible in west africa. In *West African Information and Communications Technology Congress (WAFICT)*.

Kaufman, J. (1981). Blocking in a shared resource environment. *IEEE Transactions on Communications*, 1474–1481.

Kelly, F. P. (1979). *Reversibility and Stochastic Networks*. New York: John Wiley and Sons.

Kleinrock, L. (1967). Time-shared systems: A theoretical treatment. *J. of ACM*, *14*(2), 242–261.

Koukoulidis, V. (1993). *A characterization of reversible Markov processes with applications to shared-resource environments.* Ph.D. dissertation, Concordia University Montreal, Canada.

Lakshmikantha, A., Srikant, R., & Beck, C. (2010). Differential equation models of flow-size based priorities in internet routers. *International Journal of Systems, Control and Communications*, *2*(1), 170–196.

Lassila, P., Penttinen, A., & Virtamo, J. (2008). Dimensioning of data networks: a flow-level perspective. *European Transactions on Telecommunications*, *20*(6), 549–563.

Lee, J., Hespanha, J. P., Bohacek, S., & Obraczka, K. (2005). A study of TCP fairness in high-speed networks. Technical report, University of Southern California.

Lindberger, K. (1999). Balancing quality of service, pricing and utilisation in multiservice networks with stream and elastic traffic. In *ITC-16: International Teletraffic Congress*, (pp. 1127–1136).

Litjens, R., van den Berg, H., & Boucherie, R. J. (2008). Throughputs in processor sharing models for integrated stream and elastic traffic. *Performance Evaluation*, *65*, 152–180.

Logothetis, M. D. & Moscholios, I. D. (2007). Call-level multi-rate teletraffic loss models. In *International Conference on Internet Monitoring and Protection (ICIMP'2007)*, Silicon Valley, USA. Tutorial presentation.

Lundh, P., Nádas, S., Nagy, Z., & Rácz, S. (2008). A method for improved congestion detection and control in a wireless telecommunications systems. Patent application.

Metro Ethernet Forum (2011). Microwave Technologies for Carrier Ethernet Services.

Mo, J. & Walrand, J. (2000). Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, *8*(5), 556 – 567.

Nádas, S., Rácz, S., Nagy, Z., & Molnár, S. (2007). Providing congestion control in the Iub Transport Network for HSDPA. In *Globecom*, (pp. 5293–5297).

Nadiv, R. & Rosenhouse, T. (2010). Flex your backhaul network with adaptive coding & modulation. Ceragon white paper, `http://www.ceragon.com/files/ceragon_adaptive_coding_modulation_white_paper.pdf.pdf`.

Norlund, K., Ottosson, T., & Brunstrom, A. (2004). Fairness measures for best effort traffic in wireless networks. In *Proceedings of the* 15$^{\text{th}}$ *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications.*

Parkvall, S., Englund, E., Malm, P., Hedberg, T., Persson, M., & Peisa, J. (2003). WCDMA evolved – high-speed packet-data services. *Ericsson Review*, (2).

Parkvall, S., Peisa, J., Torsner, J., Sågfors, M., & Malm, P. (2005). Wcdma enhanced uplink – principles and basic operation. In *VTC 2005-Spring*, (pp. 1411–1415).

Rácz, S., Gerő, B. P., & Fodor, G. (2002). Flow level performance analysis of a multi-service system supporting elastic and adaptive services. *Performance-Evaluation*, *49*, 451–469.

Rácz, S., Telek, M., & Fodor, G. (2001). Call level performance analysis of 3rd generation mobile core networks. *ICC*.

Ramakrishnan, K., Floyd, S., & Black, D. (2001). The Addition of Explicit Congestion Notification (ECN) to IP. RFC 3168 (Proposed Standard).

Rényi, A. (1960). On measures of entropy and information. In *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability*, (pp. 547–561).

Riedl, A., Bauschert, T., & Frings, J. (2002). A framework for multi-service IP network planning. In *International Telecommunication Network Strategy and Planning Symposium (Networks)*, (pp. 183–190).

Roberts, J. (2004). A survey of statistical bandwidth sharing. *Computer Networks*, *45*(3).

Schulze, H. & Mochalski, K. (2009). Internet study 2008/2009. Technical report, ipoque GmbH.

Sprenkels, R., Pras, A., van Beijnum, B.-J., & de Goede, L. (2000). A customer service management architecture for the Internet. In A. Ambler, S. Calo, & G. Kar (Eds.), *Services Management in Intelligent Networks*, volume 1960 of *Lecture Notes in Computer Science* (pp. 71–82). Springer Berlin / Heidelberg. 10.1007/3-540-44460-2_7.

Srivastava, V. & Motani, M. (2005). Cross-layer design: a survey and the road ahead. *IEEE Communications Magazine*, *43*(12), 112–119.

Stewart, W. J. (1994). *Introduction to the Numerical Solution of Markov Chains*. Princeton, New Jersey: Princeton University Press.

Vulkán, C. & Nagy, Z. (2009). Iub/Iur HSDPA Congestion Control. In *ICT-MobileSummit 2009*.

Weigle, M. C., Jeffay, K., & Smith, F. D. (2005). Delay-based early congestion detection and adaptation: Impact on web performance. *Computer Communications*, *28*(8), 837–850.

Welzl, M. & Mühlhäuser, M. (2003). CAVT: a congestion avoidance visualization tool. *SIGCOMM Comput. Commun. Rev.*, *33*(3), 95–101.

Zukerman, M., Tan, L., Wang, H., & Ouveysi, I. (2005). Efficiency-fairness tradeoff in telecommunications networks. *IEEE Communications Letters*, *9*(7).

Publications

Book chapters

[B1] Sz. Nádas, S. Rácz and **P. Pályi**. HSPA Transport Network Layer Congestion Control. In *HSDPA/HSUPA Handbook*. CRC Press, Taylor & Francis Group, 2010. pp. 297–330.

Journal papers

[J1] B. Gerő, **P. Pályi** and S. Rácz. Flow level performance analysis of a multi-rate system supporting stream and elastic services. In *International Journal of Communication Systems*, 2012. doi: 10.1002/dac.1383.

[J2] **P. Pályi**, A. Körösi, B. Székely, J. Bíró, S. Rácz. Characterization of Peak-Rate Limited Bandwidth-Efficient Discriminatory Processor Sharing. in *Acta Polytechnica Hungarica*, 2012. pp. 151–164.

[J3] **P. Pályi**, M. Horváth and S. Rácz. RLC-alapú HSDPA szállítóhálózati torlódásvezérlés. In *Híradástechnika*, 2010. pp. 15–20.

Conference papers

[C1] **P. Pályi**, S. Rácz and Sz. Nádas. Window-based HSDPA Transport Network Congestion Control. In *Proc., European Wireless 2010*, Lucca, Italy, April 2010, pp. 123–131.

[C2] **P. Pályi**, S. Rácz and Sz. Nádas. Fairness-Optimal Initial Shaping Rate for HSDPA Transport Network Congestion Control. In *Proc., IEEE ICCS 2008*, Guangzhou, China, November 2008, pp. 1415–1421.

[C3] N. Ukic, M. Zemljic, I. Markota, **P. Pályi**, D. Asztalos. Evaluation of Bridge-Point Model-Driven Development Tool in Distributed Environment. In *Proc., SoftCOM 2011*, Split, Croatia, September 2011.

[C4] **P. Pályi**, S. Molnár. Fairness Study of HSDPA in the Transport Network. In *Proc., TRANSCOM 2007*, Zilina, Slovak Republic, June 2007, pp. 197–200.

[C5] **P. Pályi**. Application of elastic models in HSDPA transport network dimensioning. In *Proc., Poster 2007*, Prague, Czech Republic, May 2007.

[C6] **P. Pályi**. HSDPA fairness analysis in the transport network. In *Proc., HTE-BME 2007 students conference*, Budapest, Hungary, May 2007.

Patents

[P1] **P. Pályi**, S. Rácz, Sz. Nádas and Z. Nagy. Method For Achieving an Optimal Shaping Rate For a New Packet Flow. Patent application, Filing Number: PCT/SE2009/050726, 2009.

[P2] S. Rácz, Sz. Nádas and **P. Pályi**. Delayed Flow Control Action in Transport Network Layer WCDMA Communications. Patent Application, U.S. Application Serial No. 12/730,752, 2010.

[P3] **P. Pályi**, M. Skarve, S. Rácz and Sz. Nádas. Non-congestive loss in HSPA congestion control. Patent Application, Filing Number: 13162136.9, 2013.

Other publications

[O1] **P. Pályi**. RLC-based HSDPA transport network congestion control. Poster presentation in *High Speed Networks Laboratory Workshop 2010*, Balatonkenese, Hungary, May 2010.

[O2] **P. Pályi**, S. Rácz, Sz. Nádas. Fairness-Optimal Initial Shaping Rate for HSDPA Transport Network Congestion Control Presentation in *High Speed Networks Laboratory Workshop 2009*, Balatonkenese, Hungary, May 2009.