Helsinki University of Technology Networking Laboratory
Teknillinen korkeakoulu Tietoverkkolaboratorio
Espoo 2001 Report 3/2001

METHODS FOR PERFORMANCE EVALUATION OF NETWORKS: FAST SIMULATION OF LOSS SYSTEMS AND ANALYSIS OF INTERNET CONGESTION CONTROL

Pasi Lassila

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission for public examination and debate in Auditorium S4 at Helsinki University of Technology (Espoo, Finland) on the 12th of October, 2001, at 12 o'clock noon.

Helsinki University of Technology
Department of Electrical and Communications Engineering
Networking Laboratory

Teknillinen korkeakoulu
Sähkö- ja tietoliikennetekniikan osasto
Tietoverkkolaboratorio

Distribution:
Helsinki University of Technology
Networking Laboratory
P.O. Box 3000
FIN-02015 HUT
Tel. +358-9-451 2461
Fax. +358-9-451 2474
E-mail: arja.hanninen@hut.fi

ISBN 951-22-5637-1
ISSN 1458-0322

Otamedia Oy
Espoo 2001

HELSINKI UNIVERSITY OF TECHNOLOGY
P.O. BOX 1000, FIN-02015 HUT
http://www.hut.fi

ABSTRACT OF DOCTORAL DISSERTATION

Author Pasi Lassila

Name of the dissertation

Methods for Network Performance Evaluation: Fast Simulation of Loss Systems and Analysis of Internet Congestion Control

Date of manuscript September 6, 2001 | Date of the dissertation October 12, 2001

☐ Monograph ☑ Article dissertation (summary + original articles)

Department Department of Electrical and Communications Engineering

Laboratory Networking Laboratory

Field of research teletraffic theory, performance evaluation of communication networks

Opponent(s) Professor Michel Mandjes (University of Twente)

Supervisor Professor Jorma Virtamo (Helsinki University of Technology)

(Instructor)

Abstract

Performance evaluation of modern telecommunication networks by means of mathematical modeling frequently results in a situation, whereby an exact analytical solution poses a difficult problem in terms of computational evaluation. In this thesis, two such problems are studied and a different approach for easing the computational burden is developed in each case.

The first part of the thesis considers the problem of evaluating blocking probabilities in loss systems, which are often used as models for the call scale behavior of modern networks. In this case, the solution to the problem can be given a well known analytical expression, but in practice it can not be used for computing the blocking probabilities due to the prohibitive size of the state space of the system. Then one can use simulation to obtain estimates of the blocking probabilities. For increasing the efficiency of the simulation, i.e., for reducing the variance of the estimates, several novel and increasingly more efficient methods are presented. Noticeable variance reductions are obtained by applying the method of conditional expectations. However, even greater variance reductions are gained by using importance sampling. In the thesis several importance sampling based methods are presented, of which the inverse convolution approach provides variance reductions surpassing all previously reported results in the literature.

In the second part of the thesis, the problem of congestion control in the Internet is studied. Specifically, the focus is on modeling the interaction between the TCP rate control algorithm and the RED buffer management mechanism. By using various analytical approximations, a novel dynamic model is derived for describing the interaction between an idealized TCP source population and a RED controlled buffer. Ultimately, the model consists of a set of coupled retarded functional differential equations (RFDEs) governing the time dependent expectations of the stochastic system state variables. This model is used to explore the dependency of the equilibrium of the system on the parameters of the physical system. Additionally, methods are derived allowing the stability of the system to be analyzed. In particular, sufficient and necessary conditions are obtained for the RFDE system, such that the system is asymptotically stable.

Keywords MC methods, importance sampling, loss systems, TCP modeling, RED, congestion control, stability

UDC	621.395: 004.7: 519.2	Number of pages	86 p. + app. 108 p.
ISBN (printed)	951-22-5632-0	ISBN (pdf)	951-22-5637-1
ISBN (others)		ISSN	1458-0322

Publisher Networking Laboratory/Helsinki University of Technology

Print distribution

☑ The dissertation can be read at http://www.hut.fi/Yksikot/Kirjasto/Diss/

Preface

This thesis contains the results from the research I have been involved in from the beginning of 1997 until the spring of 2001 while working in the Networking Laboratory at the Helsinki University of Technology. The work has been done as part of the projects COST257 and COM^2, for which the funding has been provided by Tekes, Sonera, Nokia Telecommunications, and the Finnish Academy. Additionally, I have received funding in the form of grants from the Wihuri Foundation and the Nokia Foundation.

I have had the privilege of having Professor Jorma Virtamo as my supervisor throughout my Ph.D. studies. Without his invaluable insight, dedication to high quality in his research and active involvement in the work, this thesis would not have been possible, and for that I am deeply grateful to him. Additionally, I wish to express my gratitude to my colleagues with whom I have had the pleasure of working with, Ph.D. Samuli Aalto, Mr. Jouni Karvo (Lic. Tech.), and Dr. Pirkko Kuusela, for their encouragement and considerable input in the work. The careful reading of the final manuscript and the helpful comments provided by the examiners of this thesis, Professor Ilkka Norros and Dr. Poul Heegaard, have been greatly appreciated.

The staff at the laboratory deserve their share of the thanks, as well. The relaxed and friendly atmosphere of the lab has helped in making the notion of going to work seem less serious. Also, big Thank Yous are in order to my parents and my family, and my "second family" at Vilppulantie for their love and support.

Finally, I dedicate this thesis to the ladies of my life - Minna and my daughter Sanni.

Pasi Lassila

Espoo, August 31, 2001

Contents

List of Publications

1. P. Lassila, J. Virtamo, "Using Gibbs Sampler in Simulating Multiservice Loss Systems", in Proceedings of IFIP TC6/WG6.3 7^{th} International Conference on Performance of Information and Communications Systems, PICS '98, Lund, Sweden, May 1998, pp. 261–272.

2. P. Lassila, J. Virtamo, "Variance Reduction in Monte Carlo Simulation of Product Form Systems", IEE Electronics Letters, vol. 34, no. 12, 1998, pp. 1204–1205.

3. P. Lassila, J. Virtamo, "Efficient Importance Sampling for Monte Carlo Simulation of Loss Systems", in Proceedings of the 16^{th} International Teletraffic Congress, ITC 16, Edinburgh, United Kingdom, June 1999, pp. 787-796.

4. P. Lassila, J. Virtamo, "Nearly Optimal Importance Sampling for Monte Carlo Simulation of Loss Systems", ACM Transactions on Modeling and Computer Simulation, vol. 10, no. 4, October 2000, pp. 326–347.

5. P. Lassila, J. Karvo, J. Virtamo, "Efficient Importance Sampling for Monte Carlo Simulation of Multicast Networks", in Proceedings of IEEE INFOCOM 2001, Anchorage, USA, April 2001, pp. 432–439.

6. P. Lassila, J. Virtamo, "Modeling the Dynamics of the RED Algorithm", in Proceedings of the 1^{st} COST 263 International Workshop: Quality of Future Internet Services, QofIS 2000, Berlin, Germany, September 2000, pp. 28–42.

7. P. Kuusela, P. Lassila, J. Virtamo, P. Key, "Modeling RED with Idealized TCP Sources", in Proceedings of the 9^{th} IFIP Working Conference on Performance Modeling and Evaluation of ATM & IP Networks, IFIP ATM & IP 2001, Budapest, Hungary, June 2001, pp. 155–166.

8. P. Kuusela, P. Lassila, J. Virtamo, "Stability of TCP-RED Congestion Control", in Proceedings of the 17^{th} International Teletraffic Congress, ITC 17, Bahia de Salvador, Brazil, September 2001, pp. 655–666.

Chapter 1

Introduction

Performance analysis of modern communication networks is a complex task as the networks contain phenomena on time scales, which can differ by several orders of magnitude. For example, the time scale of cell or packet arrivals into a buffer is microseconds and the time scale of call or session arrivals is seconds or fractions of a second. Modeling all time scales with a single model would be impossible. Typically, the problem of time scales is handled by assuming that the processes on the lower time scales are quasi-stationary with respect to the higher time scales, allowing us to decouple the different time scales from each other (see, e.g., Hui [16]). In the context of ATM, the division into time scales resulted in the use of so called cell, burst, and call scale models, which have been extensively studied, e.g., in Roberts et al. [30].

However, given that one can apply the principle of separation of time scales in modeling to reduce model complexity, it is still often the case that a precise and accurate model with known analytical results poses difficult problems in terms of the computability of the performance measures of interest. Thus, approximative models or methods, such as simulation methods, need to be developed to obtain approximations or estimates of the performance measures. This is particularly important when the model is used as a basis for performing network dimensioning (network optimization) or dynamic network control. Then, it is necessary to attain a good balance between two sometimes conflicting criteria: model accuracy and computability. In this work, the emphasis is on devising models and methods that enable us quickly and with a sufficient degree of accuracy to compute certain performance measures. Two different problems have been considered and a different approach has been applied in each case.

In the first case, the use of efficient simulation techniques to obtain estimates of call blocking probabilities in loss systems is studied. In particular, two loss systems are considered: the multiservice loss system, which has been studied in detail, e.g., by Ross in [31], and the multicast loss system, as studied by Karvo et al. in [17] and Nyberg et al. in [27]. In general, loss systems are mathematically well understood and analytical expressions for the blocking probabilities can be easily written down. However, a problem with the exact solution is the prohibitive size of the state space, which renders a direct calculation computationally

intractable.

In such a case, one can use simulation to estimate the blocking probabilities to the desired level of accuracy. For the loss systems considered here, the form of the stationary distribution is known, and static Monte Carlo (MC) simulation techniques can be applied for estimating the blocking probabilities. To increase the efficiency of the simulation, i.e., to reduce the variance of the estimator, there are several alternatives available in the literature (see, e.g., [32, chap. 4]).

In this work, the methods of conditional expectations and importance sampling (IS) are applied to obtain novel improved methods of estimating the blocking probabilities. The published methods represent increasingly efficient solutions to the problem. The first results in variance reduction techniques can be found in Publications 1 and 2, where the method of conditional expectations is applied. The results in Publication 3 represent a substantial step forward in terms of efficiency. In the publication, only IS is applied and a new composite form IS distribution is defined. Publications 4 and 5 contain the most significant results, obtained with using only IS. In both publications, a decomposition property of the system is utilized, allowing the breaking down of the original problem into independent subproblems. To solve each subproblem, an appropriate conditional IS distribution is defined and a novel algorithm has been given to generate the samples efficiently. In terms of the variance reductions obtained with the method, it surpasses all previously reported methods in the literature.

Instead of using simulations to estimate performance measures, one can develop analytical approximations to ease the burden of computing performance measures. This approach is used in the second part of this work. The problem that is investigated deals with Internet congestion control. In this context, the IETF (Internet Engineering Task Force) has recently defined in their standards Active Queue Management (AQM) methods that aim to increase fairness among the flows sharing a congested buffer and to alleviate such problems as the global synchronization of TCP flows, see [42] (RFC2309). One of the most prominent of the AQM methods is the Random Early Discard (RED) algorithm proposed by Floyd and Jacobson in [50]. The RED algorithm was designed to work in cooperation with the TCP sources such that when congestion starts building up in the buffer (i.e., the load increases), the RED algorithm begins to drop packets randomly, causing some of the TCP sources to reduce their sending rates.

The aim in the study of the RED algorithm is to create an analytical dynamic model for the interaction of the TCP source population and the RED controlled queue. An additional goal is that the model should be simple enough to be used for exploring the impact of various system parameters on the stability of the system.

The system can be considered as consisting of two parts: 1) a queue receiving an aggregate packet arrival stream generated by a population of TCP sources, and 2) the population of TCP sources reacting dynamically to successful packet transmissions and random packet losses according to the TCP flow control algorithm. The focus is first on part 1 of the problem. As studied by Sharma et al. in [71], in modeling the system, even under the simplest

assumptions regarding the arrival process and packet length distributions, one obtains a problem for which the direct computation of the performance measures is impossible. Thus, again a situation is encountered in which exact analytical models and formulas are impractical from a computational perspective, and approximative methods are required. First results in this direction were reported in [71]. Most importantly, it introduces the approach to modeling the time dependent behavior of the system in the "mean" sense by using differential equations. This approach is continued in Publication 6, where a model describing the complete queue dynamics is presented. This model enables us to apply control theoretic principles for studying the behavior of the system. It is possible, for example, to define criteria in terms of the parameters of the system such that, upon a load change, the system reaches the new equilibrium in a controlled manner. In Publication 7, a model is presented that closes the control loop from the queue back to the TCP sources, i.e., a model for parts 1 and 2 above. The model captures the dynamics of the interaction between an idealized TCP source population and the RED controlled queue. Finally, the stability of the system developed in Publication 7 is analyzed in Publication 8.

The thesis is organized as follows. The two problems that have been studied are treated separately in the two subsequent chapters. Under each chapter, the contributions of the work have been summarized in a consistent manner using the same notation throughout (which is not necessarily the case in the publications). In Chapter 4, a summary of each publication is presented, along with comments on the author's contributions.

Chapter 2

Fast Simulation Techniques for Loss Systems

2.1 Introduction

In general, loss systems can be used to model a situation where users establish calls through a network and associated with each call are the resource (bandwidth) requirements on the links along the route of the call. Here the term call is used in a sense which does not necessarily imply the presence of a circuit switched network with advance resource reservations, but it can also represent a flow in a packet switched network having a certain (fixed) bandwidth requirement along the route of the flow. When the call is offered but there is not enough bandwidth on all the links along the requested route, the call is blocked and lost. In this work, two loss systems have been considered: the multiservice loss system, which has been studied in detail, e.g., by Ross in [31], and the multicast loss system, as studied by Karvo et al. in [17] and by Nyberg et al. in [27].

Under certain assumptions, the steady state distribution of the state of the system, the number of calls in each class, identified by the route and the bandwidth requirements of the call, has the well known product form. One of the most important performance measures in these systems is the blocking probability of a call, for which an exact analytical expression also exists. However, associated with the exact solution is a problem of computability, namely that the state space grows exponentially with the number of traffic classes; see Kelly et al. [24] for a complexity theoretic treatment of the problem in the case of algorithms in multiservice loss systems. Recursive methods can be used to alleviate the problem (see, e.g., Kaufman [19], Roberts [29], Ross [31], Nyberg [27]) but they are applicable only in the case of a small number of links and/or special network topologies.

One approach to the problem is to derive analytical approximations for the blocking probabilities. In the case of the multiservice loss system, the analytical approaches are based either on using the so called reduced load approximation, which leads to a set of fixed point equations (see, e.g., Ross and Chung [3], Kelly [21], Mitra et al. [26]), or on mathematical

techniques applied to the generating function of the link occupancies (Choudhury et al. [2], Down and Virtamo [6] and Simonian et al. [35]). Reduced load approximations for the multicast loss system have been given by Karvo et al. in [18].

An alternative to deriving approximations is to simulate the system to a desired level of accuracy. In this section of the thesis several methods are given for the efficient estimation of the blocking probabilities by various simulation techniques. The methods have been published in Publications 1-5 and they represent the progressively increasing improvements that have been obtained during the course of the work while tackling the problem.

Traditionally, the simulation approaches have been based on either static Monte Carlo (MC) techniques or process simulation techniques (using the terminology of [23, chap. 1]). Process simulation is a method where the temporal stochastic behavior of the system under study is simulated. For example, in queuing systems this translates to a discrete event simulation of the arrival and departure of customers to/from the queue. On the other hand, static MC is by nature a method, where the behavior of the system is not simulated in time. In the case of the multiservice loss system, one way of performing process simulation is to make the assumption that the call holding times are exponentially distributed. Then the process determining the time evolution of the number of calls present in the system is described by a Markov chain, and its simulation produces positively correlated samples. However, in this work the focus has been on using static MC methods. This is possible, because the stationary distribution of the system is known, and it is feasible to, e.g., generate independent samples directly from the distribution to minimize the effect of positive correlations between the samples. Nonetheless, this standard static MC still suffers from the problem that the blocking events are (relatively) rare, and, hence, the simulation is not very efficient.

2.1.1 Review of Efficient Simulation Methods for Queuing Systems

To increase the efficiency of the simulation, several well known general variance reduction methods exist: control variables, antithetic variates, conditional expectations and importance sampling (see, e.g., [32, chap. 4] for general reviews on these techniques) and, more lately in the context of process simulation, the so called RESTART method (or splitting method). In the context of simulating queuing systems, importance sampling and RESTART have been the most commonly used variance reduction techniques.

Typically the efficiency of the proposed methods have been theoretically treated by considering a so called rare event estimation problem. Then the problem is that to estimate the probability of the rare event, say p, the number of samples $N(p)$ required to reach a fixed relative deviation, defined as the ratio of the deviation of the estimator to its expectation, by using normal simulation is roughly proportional to $1/p$ (assuming i.i.d. samples). Thus, when $p \to 0$, the required number of samples $N(p)$ to reach a fixed relative deviation is not bounded. In this asymptotic setting, when $p \to 0$ at an exponential rate, also $N(p)$ grows at an exponential rate, and a method is defined as asymptotically optimal if the number of

required samples grows at a rate that is less than any exponential [15]. In the context of importance sampling methods for queuing systems, the theory of large deviations has been used to identify IS distributions satisfying the criteria of asymptotic optimality (see, e.g., Heidelberger [15] for a survey or Cottrell et al. [4], Frater et al. [8], Parekh and Walrand [28] or Sadowsky [34] for individual results). Loosely speaking, the results in these papers are based on applying large deviations theory to a rare event problem satisfying a so called large deviations principle. As a result, one is able find out analytically the form and the parameters of the IS distribution. However, the problem with this approach is that a) it requires a considerable amount of prior knowledge of the system under study, b) the numerical solution of the parameters for the IS distribution may not be feasible and, perhaps most importantly, c) not all rare event problems satisfy a large deviation principle, and these techniques cannot even be used. From the point of view of the work in this thesis, an important paper is by Sadowsky and Bucklew [33], where the form of the asymptotically optimal IS distribution for estimating the probability of multidimensional sets has been derived to be of the composite type (in certain cases).

The use of process simulation to estimate steady state performance measures via the regenerative simulation method has been studied, e.g., by Goyal et al. in [12] (and in a number of other papers from the same author). There it has been shown that the IS method should, in this case, be dynamic: During one simulation cycle the process must be quickly driven to a state where the rare event happens and after that IS should be turned off to allow the system to "regenerate" again. In this setting, large deviation techniques could be applied to solve the problem of identifying the asymptotically optimal IS distribution for estimating the most probable way that the process reaches the rare event.

A practical approach for identifying suitable IS distributions has been proposed by Devetsikiokis and Townsend in [5] (and in a number of other papers from the same authors) that avoids the problem of determining analytically suitable parameters for the IS distribution(s). Instead, the idea is to use well known stochastic optimization algorithms (for a review on these, see, e.g., Glynn [11]) to search for the optimal parameter values for the IS distribution(s) to be used during the simulation. A simulation under this scheme consists first of a series of short simulations to identify the optimal values of the IS parameters, and, once the stochastic optimization algorithm has converged, a longer simulation run is made using these optimal values to derive the final estimate.

As has been noted, the effectiveness of importance sampling depends critically on the ability to find the right IS distribution. The RESTART method, presented by Villén–Altamirano and Villén–Altamirano in [37], is based on a known method called splitting, which has been used, e.g., in particle transmission simulation (e.g., [13, chap. 8.2]). The idea there is to estimate the probability of a rare event A by using conditioning on some less rare event C. Then we have by conditioning that $\mathrm{P}\{A\} = \mathrm{P}\{A \,|\, C\}\mathrm{P}\{C\}$. Using simulation it is easy to estimate the probability $\mathrm{P}\{C\}$ since it is estimated from the whole simulation data and it is less rare. However, the conditional probability $\mathrm{P}\{A \mid C\}$ is estimated only from the portion of the simulation where the event C occurred. Since the occurrence of event C is still relatively infrequent (although much more frequent than the occurrence of event A), the

conditional probability $\mathrm{P}\{A|C\}$ does not get estimated very well. In RESTART the idea is to increase the accuracy of the estimate for the conditional probability $\mathrm{P}\{A|C\}$ by making several independent replications of those parts of the process where the event C occurred, i.e., the process evolution is split into several paths when the threshold has been reached. RESTART can also be used with multiple conditioning events. The decision parameters in RESTART are then the conditioning events to be used and the number of sub-paths to be generated. Glasserman et al. [9] consider the parameterization problem and provide results for these decision parameters such that the method is asymptotically optimal. Akyamaç et al. [1] have presented a related method, the direct probability redistribution (DPR) method. Their method is related to RESTART in the sense that in DPR the state space is also divided into progressively rarer subsets but, unlike in RESTART, the sets do not have to be nested.

2.1.2 Review of Simulation Methods for Loss Systems

The literature dealing specifically on the simulation of loss systems is not as vast as for queuing systems. In process simulation context, Heegaard [14] has used importance sampling successfully in a regenerative simulation of the underlying Markov chain of the multiservice loss system to estimate the blocking probabilities. However, as mentioned earlier, in this thesis the emphasis is on the use of static MC methods, since the steady state distribution of the system is known.

The use of static MC with importance sampling has been studied by Ross in [31, chap. 6] and by Mandjes in [25]. They have considered the use of importance sampling distributions belonging in the family of so called exponentially twisted distributions. Ross has presented heuristics which attempt to increase the likelihood of the blocking states while, at the same, trying to limit the likelihood of generating misses from the allowed state space, resulting in a rather conservative twist. Mandjes proposes the use of an importance sampling distribution which twists the mean of the sampling distribution to match the most probable blocking state. For the efficient simulation of the multicast loss system no prior literature exists as far as the author is aware.

A related problem has been studied by Fleming et al. in [7], where the system is a loss system used to model a cell in a cellular network which is divided into a number of sectors sharing a pool of common speech channels. The model differs from the multiservice loss system only in how the state space boundaries are defined. The paper uses control variates to obtain variance reductions. The basic idea is to use as control variables the same variables that are to be estimated in the system under study, but corresponding to a simplified system for which the value of the control variable can be computed analytically. This approach could also be used in estimating the blocking probabilities in the multiservice loss system, but it has not been pursued in the course of this study.

2.1.3 Overview of the Chapter

In this chapter a summary is presented of the results obtained by applying the methods of conditional expectations and importance sampling in the multiservice loss system and the multicast loss system. To this end, some basic notation is required. The notation will overlap in the two systems somewhat as both systems share many properties and concepts but they may have different definitions in each of the systems. Thus, the two loss systems are treated separately in the following two sections (Section 2.2 and 2.3), and the results related to each particular loss system are summarized under the section in question.

2.2 Efficient Simulation of Multiservice Loss Systems

2.2.1 The Multiservice Loss System

Consider a network consisting of J links, indexed with $j = 1, \ldots, J$, with link j having a capacity of C_j resource units. The network supports K classes of calls. Calls of class-k arrive according to a Poisson process with arrival rate λ_k. New calls are always accepted if there is enough capacity and the blocked calls are cleared. The call holding times are exponentially distributed with mean μ_k. Associated with a class-k call, $k = 1, \ldots, K$, is an offered load $\rho_k = \lambda_k/\mu_k$ and a bandwidth requirement of b_k^junits on link j, i.e., b_k^j are assumed to be integer multiples of a basic bandwidth unit. Also, $b_k^j = 0$ when a class-k call does not use link j. Finally, the vector $\mathbf{b}^j = (b_1^j, \ldots, b_K^j)$ denotes the required bandwidths of different classes on link j.

Let us first assume that the system has infinite link capacities and let $\mathbf{X} = (X_1, \ldots, X_K)$ denote the state of the infinite capacity system, with X_k giving the number of class-k calls in progress. Then the system behaves as K independent Poisson processes and the state space is

$$\mathcal{I} = \{\mathbf{x} \mid \mathbf{x} \geq \mathbf{0}\},$$

where $x_k \in \mathbb{N}, k = 1, \ldots, K$, with $\mathbb{N}$ denoting the set of natural numbers $\{0, 1, 2, \ldots\}$. The steady state distribution of $\mathbf{X}$ is of the product form

$$p(\mathbf{x}) = \mathrm{P}\{\mathbf{X} = \mathbf{x}\} = \prod_{k=1}^{K} p_k(x_k), \quad \mathbf{x} \in \mathcal{I}, \tag{2.1}$$

where $p_k(x) = (\rho_k^x/x!)\, e^{-\rho_k}$ denotes the point probabilities of the one-dimensional Poisson distribution. Note that instead of having the infinite space $\mathcal{I}$ any Cartesian product space could be considered.

For the finite capacity system, the set of allowed states, $\mathcal{S}$, consists of those states for which the resulting link occupancies of all the links in the network do not exceed the capacity limit of any link. Formally $\mathcal{S}$ is defined as

$$\mathcal{S} = \left\{\mathbf{x} \in \mathcal{I} \mid \forall\, j : \mathbf{b}^j \cdot \mathbf{x} \leq C_j\right\},$$

where the scalar product is defined as $\mathbf{b}^j \cdot \mathbf{x} = \sum_k b_k^j x_k$. The process describing the number of calls in each class is a reversible Markov process satisfying the detailed balance equations. Thus, the steady state distribution, $\pi(\mathbf{x})$, of the process $\tilde{\mathbf{X}}$ defined in the finite state space $\mathcal{S}$ is given by the truncation of (2.1) to the allowed state space, $\mathcal{S}$,

$$\pi(\mathbf{x}) = \mathrm{P}\{\tilde{\mathbf{X}} = \mathbf{x}\} = \mathrm{P}\{\mathbf{X} = \mathbf{x} | \mathbf{X} \in \mathcal{S}\} = \begin{cases} \dfrac{\mathrm{P}\{\mathbf{X} = \mathbf{x}\}}{\mathrm{P}\{\mathbf{X} \in \mathcal{S}\}}, & \mathbf{x} \in \mathcal{S}, \\ 0, & \text{otherwise.} \end{cases} \tag{2.2}$$

Additionally, it can be noted that the process $\tilde{\mathbf{X}}$ has the steady state distribution (2.2) also when the call holding times have a general distribution with mean μ_k due to the well known insensitivity result (see, e.g., [31, p. 163]).

The set of blocking states for a class-k call, $\mathcal{B}_k$, consists of those states for which an addition of one more call from class k to a given state results in a link occupancy on some link violating the link capacity constraint,

$$\mathcal{B}_k = \left\{ \mathbf{x} \in \mathcal{S} \mid \exists\, j : \mathbf{b}^j \cdot (\mathbf{x} + \mathbf{e}_k) > C_j \right\},$$

where $\mathbf{e}_k$ is a K-component vector with 1 in the k^{th} component and zeros elsewhere. Also, let $\mathcal{R}_k$ denote the set of links that the traffic class k uses, i.e.,

$$\mathcal{R}_k = \{j \in \mathcal{J} \mid b_k^j > 0\},$$

where $\mathcal{J} = \{1, 2, \ldots, J\}$ denotes the set of link indexes. The blocking probability of a class-k call, B_k, can then be expressed in the form of a ratio of two state sums

$$B_k = \mathrm{P}\{\tilde{\mathbf{X}} \in \mathcal{B}_k\} = \sum_{\mathbf{x} \in \mathcal{B}_k} \pi(\mathbf{x}) = \frac{\sum_{\mathbf{x} \in \mathcal{B}_k} p(\mathbf{x})}{\sum_{\mathbf{x} \in \mathcal{S}} p(\mathbf{x})} = \frac{\mathrm{P}\{\mathbf{X} \in \mathcal{B}_k\}}{\mathrm{P}\{\mathbf{X} \in \mathcal{S}\}} = \frac{\beta_k}{\gamma}, \tag{2.3}$$

where $\beta_k = \mathrm{P}\{\mathbf{X} \in \mathcal{B}_k\}$ and $\gamma = \mathrm{P}\{\mathbf{X} \in \mathcal{S}\}$.

The Basic Estimation Problem

The basic simulation problem can now be formulated in two ways. The simplest formulation is obtained by defining from (2.3) B_k to be an expectation

$$B_k = \mathrm{E}[1_{\tilde{\mathbf{X}} \in \mathcal{B}_k}], \tag{2.4}$$

for which the following unbiased estimator can be given

$$\hat{B}_k = \frac{1}{N} \sum_{n=1}^{N} 1_{\tilde{\mathbf{X}}_n \in \mathcal{B}_k}, \tag{2.5}$$

with N denoting the number of samples of $\tilde{\mathbf{X}} \in \mathcal{S}$.

Alternatively, by (2.3), B_k can be expressed as a ratio of two expectations

$$B_k = \frac{\mathrm{E}[1_{\mathbf{X}\in\mathcal{B}_k}]}{\mathrm{E}[1_{\mathbf{X}\in\mathcal{S}}]}, \tag{2.6}$$

and the corresponding estimator is given by

$$\hat{B}_k = \frac{(1/N)\sum_{n=1}^{N} 1_{\mathbf{X}_n\in\mathcal{B}_k}}{(1/N)\sum_{n=1}^{N} 1_{\mathbf{X}_n\in\mathcal{S}}}, \tag{2.7}$$

where N is now the number of samples of $\mathbf{X} \in \mathcal{I}$. Note that (2.7) is a biased but consistent estimator, i.e., for a finite N for (2.7), $\mathrm{E}[\hat{B}_k] \neq B_k$, but $\mathrm{E}[\hat{B}_k] \to B_k$ when $N \to \infty$.

Both formulations have been used as starting points in the publications. Publication 1 and 2 are based on using (2.4) and Publications 3–5 use (2.7).

2.2.2 Decomposition

An important finding in this work is the observation that the problem of estimating β_k, i.e., the numerator in (2.6), can readily be decomposed into independent simpler sub-problems. This property is used when developing the most efficient IS based methods for estimating the blocking probabilities in Publications 4 and 5.

First, the set $\mathcal{D}_k^j \in \mathcal{S}$ is introduced denoting the set of blocking states for link j consisting of the points

$$\mathcal{D}_k^j = \left\{\mathbf{x} \in \mathcal{I} \mid C_j - b_k^j < \mathbf{b}^j \cdot \mathbf{x} \leq C_j\right\}.$$

This set consists of the set of blocking states of class k in a system where only link j has a finite capacity and all other links have an infinite capacity. The decomposition is based on the following observation. The set of blocking states (for traffic class k) can be expressed as

$$\mathcal{B}_k = \mathcal{S} \cap \bigcup_{j\in\mathcal{R}_k} \mathcal{D}_k^j.$$

This is illustrated in Figure 2.1 on the left hand side, which shows a two traffic class example with three links. The grey areas represent the blocking state regions $\mathcal{D}_k^j$ of a traffic class k for each link. The whole set of blocking states $\mathcal{B}_k$ is then the area between the solid black lines.

Now, β_k is an expectation of the form $\mathrm{E}[h_k(\mathbf{X})]$ with $h_k(\cdot)$ being the indicator function of the set $\mathcal{B}_k$. Based on the above, $h_k(\cdot)$ can be decomposed as

$$\begin{aligned} h_k(\mathbf{x}) &= 1_{\mathbf{x}\in\mathcal{B}_k} \\ &= \sum_{j\in\mathcal{R}_k} \frac{1}{\nu_k^j(\mathbf{x})} 1_{\mathbf{x}\in\mathcal{S}} 1_{\mathbf{x}\in\mathcal{D}_k^j}, \\ &= \sum_{j\in\mathcal{R}_k} h_k^j(\mathbf{x}), \end{aligned}$$

where

$$h_k^j(\mathbf{x}) = \frac{1}{\nu_k^j(\mathbf{x})}\, 1_{\mathbf{x}\in\mathcal{S}}\, 1_{\mathbf{x}\in\mathcal{D}_k^j},$$

and $\nu_k^j(\mathbf{x})$ is a function giving the number of sets $\mathcal{D}_k^j$ that the point $\mathbf{x}$ belongs to, i.e., it takes care of weighting appropriately those points that lie in the intersection of two or more sets $\mathcal{D}_k^j$.

Thus, the computation of the original expectation decomposes into independent subproblems, i.e,

$$\beta_k = \mathrm{E}[1_{\mathbf{X}\in\mathcal{B}_k}] = \sum_{j\in\mathcal{R}_k} \mathrm{E}[h_k^j(\mathbf{X})].$$

The value of one of the $h_k^j(\cdot)$ functions is illustrated in Figure 2.1 on the right hand side. Each expectation $\mathrm{E}[h_k^j(\mathbf{X})]$ can be thought of as the blocking probability contribution due to link j.

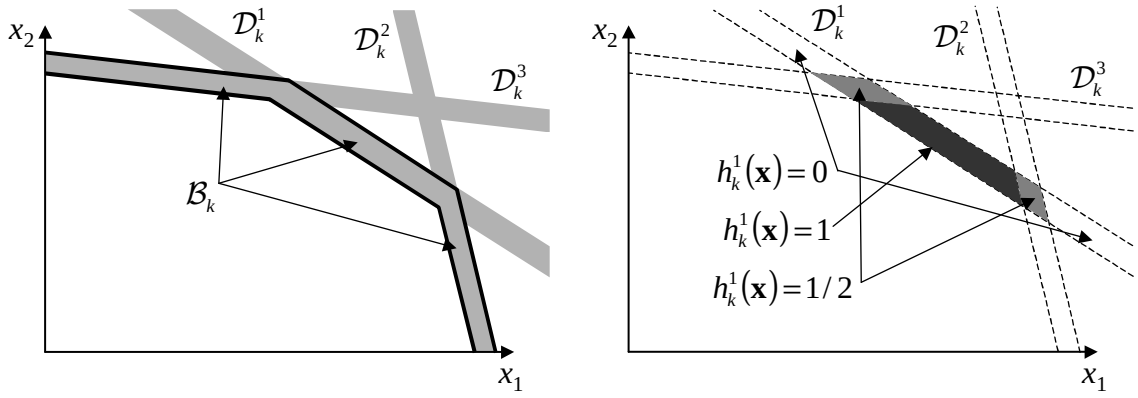

Figure 2.1: Decomposition of the set $\mathcal{B}_k$ into three subsets in a network with two traffic classes and three link constraints (left figure) and the values of one of the $h_k^j(\cdot)$ functions in different parts of $\mathcal{D}_k^j$ (right figure).

With slight modification the set $\mathcal{B}_k$ can be decomposed into non-overlapping regions, whence there is no need for the $1/\nu_k^j(\mathbf{x})$ term in the $h_k^j(\mathbf{x})$ function. This can be done by attributing each state that belongs to multiple sets $\mathcal{D}_k^j$ uniquely to only one $\mathcal{D}_k^j$. One alternative is to attribute the state to the link having the smallest link index j. This new set is denoted by $\mathcal{E}_k^j$ and it is defined as

$$\mathcal{E}_k^j = \mathcal{S} \cap \left\{ \mathbf{x} \in \mathcal{I} \,:\, \mathbf{x} \in \mathcal{D}_k^j \;\wedge \mathbf{x} \notin \mathcal{D}_k^{j'}, \forall j' < j \right\}.$$

Note that the intersection with $\mathcal{S}$ is now included in the definition of the set $\mathcal{E}_k^j$. Now, the $\mathcal{E}_k^j$ obviously form a partitioning of $\mathcal{B}_k$, i.e.,

$$\mathcal{B}_k = \bigcup_{j\in\mathcal{R}_k} \mathcal{E}_k^j,$$

and $\mathcal{E}_k^j \cap \mathcal{E}_k^{j'} = \emptyset$, when $j \neq j'$. Thus, the estimation problem of β_k can be formulated entirely in terms of probabilities, instead of expectations,

$$\beta_k = \mathrm{P}\{\mathbf{X} \in \mathcal{B}_k\} = \sum_{j \in \mathcal{R}_k} \mathrm{P}\{\mathbf{X} \in \mathcal{E}_k^j\}.$$

Here each probability $\mathrm{P}\{\mathbf{X} \in \mathcal{E}_k^j\}$ can be thought of as the blocking probability contribution due to link j.

The partition of $\mathcal{B}_k$ into non-overlapping sets $\mathcal{E}_k^j$ is illustrated in Figure 2.2 for the same two traffic class example as in Figure 2.1. In the figure, $\mathcal{B}_k$ is represented by the grey area and it is formed by the union of three disjoint subsets $\mathcal{E}_k^2, \mathcal{E}_k^3$ (light grey areas) and $\mathcal{E}_k^1$ (dark grey area). Also, note that each $\mathcal{E}_k^j$ is a subset of the corresponding $\mathcal{D}_k^j$.

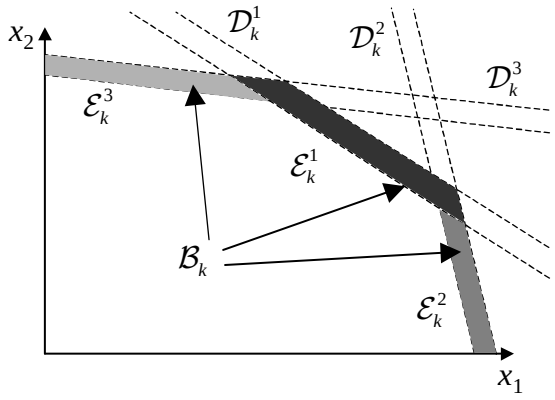

Figure 2.2: Decomposition of $\mathcal{B}_k$ into sets $\mathcal{D}_k^j$ and $\mathcal{E}_k^j$.

2.2.3 Sample Generation Methods

The efficient generation of samples having some desired distribution is an essential part of the simulation problem. In this section, two methods are considered, the rejection method and the Gibbs sampler. It is shown how they are applied in the multiservice loss system context to generate samples of $\tilde{\mathbf{X}}$, i.e., samples obeying the distribution $\pi(\mathbf{x})$ as in (2.2). Note that the methods apply more generally for generating any random variables having some distribution in a given set.

In the traditional Monte Carlo method the idea is to generate independent samples of $\mathbf{X} \in \mathcal{S}$, i.e., samples of $\tilde{\mathbf{X}}$. In the case of multiservice loss systems one often uses the rejection method which consists of generating samples of $\mathbf{X}$ in a larger space $\mathcal{I}$, where the components of $\mathbf{X}$ are independent, and rejecting those samples which fall outside the allowed state space $\mathcal{S}$. Note that the samples can actually be generated in any Cartesian product space enclosing $\mathcal{S}$. In practice, one would choose the smallest such a space, which is the space limited by the maximum number of calls allowed for each class. The accepted samples

then have the distribution $\pi(\mathbf{x})$ and (2.5) is used to estimate the blocking probabilities. In most of the work reported here, the Monte Carlo rejection method has been used.

The MC rejection method, described above, allows the generation of independent samples from a distribution. However, the estimation of the blocking probabilities by simulation does not require the samples to be independent. Positive correlation between the samples just makes the estimator less efficient from the point of view of the variance. In the Gibbs sampler the idea is to generate a Markov chain having the desired stationary distribution, by using transition probabilities based on conditioning. The states of this chain are then used as samples.

The idea is based on the following very general property (see Publication 1 or [36] for more details). Recall that $\tilde{\mathbf{X}}$ is a vector random variable with state space $\mathcal{S}$ and stationary distribution $\pi(\mathbf{x})$. Let $\mathcal{A}_1, \ldots, \mathcal{A}_I$ form a partition of the state space and let $\iota(\tilde{\mathbf{X}})$ denote the unique index of the set to which the state $\tilde{\mathbf{X}}$ belongs to. Then the Markov chain $\tilde{\mathbf{X}}_n$ with the transition probability

$$\mathrm{P}\{\tilde{\mathbf{X}}_{n+1} = \mathbf{y} \,|\, \tilde{\mathbf{X}}_n = \mathbf{x}\} = \mathrm{P}\{\tilde{\mathbf{X}} = \mathbf{y} \,|\, \tilde{\mathbf{X}} \in \mathcal{A}_{\iota(\mathbf{x})}\} \tag{2.8}$$

has the invariant distribution $\pi(\mathbf{x})$.

Let $\mathbf{P}$ denote the transition probability matrix with the components given by (2.8). The Markov chain generated by this transition matrix is reducible, because there are no transitions between different sets. However, by defining several partitions $1, \ldots, M$ with transition matrices $\mathbf{P}^{(m)}$ the Markov chain $\tilde{\mathbf{X}}_n$ corresponding to the compound transition matrix $\mathbf{P} = \mathbf{P}^{(1)} \cdots \mathbf{P}^{(M)}$ is irreducible with a suitable choice of the partitions. Then $\pi(\mathbf{x})$ will be the unique stationary distribution of the chain.

For the purpose of estimating the blocking probability of class-k calls in the multiservice loss system, the partition is defined to consist of sets in the coordinate direction of traffic class k, which leads to the so called Gibbs sampler. Considering all the traffic classes there are altogether K partitions. Let us denote by $\mathcal{A}_k^i$ the i^{th} set in partition k. Using a two traffic class example, shown in Figure 2.3 (left picture) for traffic class 2, the partition consists of the vertical columns. Each set $\mathcal{A}_k^i$ consists of states where the number of calls of all other classes are fixed, but the k^{th} component varies. In general, we refer to sets $\mathcal{A}_k^i$ as k-columns. The set of blocking states $\mathcal{B}_k$ for the class-k calls consists of the end points of the k-columns. The Markov chain $\tilde{\mathbf{X}}_n$ generated by the compound transition matrix $\mathbf{P}$ is irreducible since it is possible to move from any state $\mathbf{x}$ in the coordinate convex state space $\mathcal{S}$ to any other state $\mathbf{y}$ with at most K steps in alternating directions.

The simulation of the Markov chain $\tilde{\mathbf{X}}_n$ consists of making transitions with the transition matrices $\mathbf{P}^{(k)}$ in cyclical order. This is illustrated for the two traffic class example in Figure 2.3 (right picture). In transitions generated with $\mathbf{P}^{(k)}$ only the component x_k changes. Starting from state $\tilde{\mathbf{X}}_n$ the value of x_k of the next state is obtained by drawing it from the one-dimensional truncated Poisson distribution $p_k(x_k)/g_k(L_k(\mathbf{X}_n))$ with $x_k \in (0, \ldots, L_k(\tilde{\mathbf{X}}_n))$, $L_k(\tilde{\mathbf{X}}_n)$ denoting the length of the k-column to which the state $\tilde{\mathbf{X}}_n$

belongs, and $g_k(L)$ denoting the normalization sum

$$g_k(L) = \sum_{l=0}^{L} p_k(l) = \sum_{l=0}^{L} \frac{\rho_k^l}{l!} e^{-\rho_k}.$$

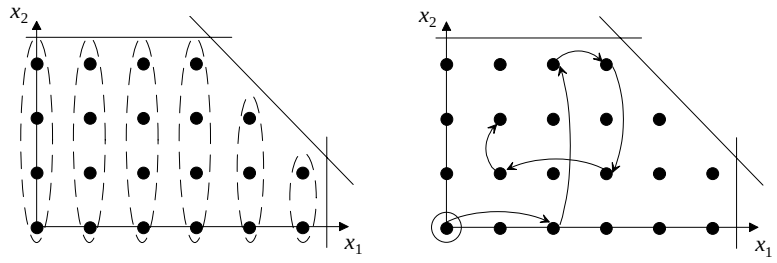

Figure 2.3: State space partitioning and Gibbs sampler example.

The Gibbs sampler provides a simple way of generating samples with the distribution $\pi(\mathbf{x})$ without any misses, which occur with the traditional Monte Carlo rejection method. All the samples $\tilde{\mathbf{X}}_n$ have directly the correct distribution $\pi(\mathbf{x})$ and can be used in estimator (2.5). This property is particularly useful when the traffic loads are high, which would cause a lot of misses to be generated in the Monte Carlo rejection method. The Gibbs sampler is used in this thesis in Publication 1 and was also applied by Lassila and Virtamo in [22].

2.2.4 Variance Reduction by Conditional Expectations

The method of conditional expectations is a well known variance reduction technique in simulations, see [32, chap. 4]. Let $\mathbf{X} \in \mathcal{S}$ denote here temporarily any random variable with some distribution P and assume that we are interested in the value of the expectation $H = \mathrm{E}[h(\mathbf{X})]$. The following elementary identity holds always,

$$H = \mathrm{E}[h(\mathbf{X})] = \mathrm{E}[\mathrm{E}\,[h(\mathbf{X}) \,|\, g(\mathbf{X})]],$$

where $g(\cdot)$ is another function. Assume now that the conditional expectation

$$\eta(\mathbf{x}) = \mathrm{E}\,[h(\mathbf{X}) \,|\, g(\mathbf{X}) = \mathbf{x}]$$

can be calculated analytically. Then the expectation of $h(\mathbf{X})$ becomes

$$H = \mathrm{E}[h(\mathbf{X})] = \mathrm{E}[\eta(g(\mathbf{X}))]. \tag{2.9}$$

More specifically, of interest here is the case where the state space $\mathcal{S}$ has a partitioning into sets $\mathcal{A}^i,\ i = 1, \ldots, I$. A state $\mathbf{X}$ belongs to one and only one of the sets $\mathcal{A}^i$. Let us denote

the unique index of this set by $\iota(\mathbf{X})$. This discrete valued function is used as the function $g(\cdot)$ in the above formulae. So, finally, the estimator is

$$\hat{H} = \frac{1}{N}\sum_{n=1}^{N}\eta(\iota(\mathbf{X}_n)) = \frac{1}{N}\sum_{i=1}^{I}\eta(i)N_i, \tag{2.10}$$

where N_i is the count of the samples having $\iota(\mathbf{X}_n) = i$ or, equivalently, $\mathbf{X}_n \in \mathcal{A}^i$, and $\eta(i) = \mathrm{E}\left[h(\mathbf{X}) \,|\, \mathbf{X} \in \mathcal{A}^i\right]$. Intuitively it is clear that estimator (2.10) has a lower variance than the basic estimator of H, i.e., $\hat{H} = (1/N)\sum_n h(\mathbf{X}_n)$, since part of the solution to the problem is computed analytically, thus eliminating some randomness from the results. In the i.i.d. case, this can also be shown easily analytically, see Publication 2. However, the real problem is actually to find a suitable conditioning function $\eta(\cdot)$ whose value can be readily computed analytically. The conditional expectations method to be described here is used in Publications 1 and 2.

Now, let $\mathbf{X}$ again denote the state variable in the multiservice loss system. Let us first assume that the Monte Carlo rejection method (or the Gibbs sampler) is used to generate samples of $\tilde{\mathbf{X}} \in \mathcal{S}$. In the case of the multiservice loss system, the same K partitions as with the Gibbs sampler can be used, i.e., partition k consists of columns in the direction k, and $\mathcal{A}_k^i$ denotes the i^{th} k-column in partition k. The blocking probability, B_k, is an expectation of the above considered type with $h(\tilde{\mathbf{X}}) = 1_{\tilde{\mathbf{X}}\in\mathcal{B}_k}$ (cf. eq. (2.4)). Because of the product form of the state probabilities (2.2) the conditional expectation $\eta_k(i) = \mathrm{E}\left[1_{\tilde{\mathbf{X}}\in\mathcal{B}_k} \,|\, \tilde{\mathbf{X}} \in \mathcal{A}_k^i\right]$ can be calculated easily. The probability distribution constrained to column i in partition k (set $\mathcal{A}_k^i$) is a truncated Poisson distribution and the conditional blocking probability is given by

$$\eta_k(i) = \mathrm{E}\left[1_{\tilde{\mathbf{X}}\in\mathcal{B}_k} \,|\, \tilde{\mathbf{X}} \in \mathcal{A}_k^i\right] = \mathrm{erl}(L_k^i, \rho_k),$$

where L_k^i denotes the length of the k-column i (set $\mathcal{A}_k^i$) and $\mathrm{erl}(L, \rho)$ denotes the well known Erlang loss formula,

$$\mathrm{erl}(L, \rho) = \frac{\rho^L/L!}{\sum_{n=1}^{L}\rho^n/n!}.$$

This leads to the estimator

$$\hat{B}_k = \frac{1}{N}\sum_{n=1}^{N}\mathrm{erl}(L_k(\tilde{\mathbf{X}}_n), \rho_k),$$

where, for clarity, $L_k(\tilde{\mathbf{X}})$ has been used for the length of the k-column to which the state $\tilde{\mathbf{X}}$ belongs, instead of $L_k^{\iota(\tilde{\mathbf{X}})}$.

In the above, the samples $\tilde{\mathbf{X}}_n$ have to be generated in the state space $\mathcal{S}$ from the distribution (2.2). However, as was shown earlier, from (2.6) the blocking probability can be defined by considering a random vector $\mathbf{X}$ in a larger state space $\mathcal{I}$. The Monte Carlo method can be applied both for the numerator and the denominator leading to the estimator (2.7). Note that if the same samples $\mathbf{X}_n$ are used in both the numerator and the denominator, this

estimator, in effect, reduces to the estimator (2.5), where in the denominator one would have the number of accepted samples from the rejection method.

Something different, however, is obtained by noting that both the numerator and the denominator of (2.6) can be estimated with the conditional expectation method. To this end, we define first the space $\tilde{\mathcal{I}}$, which is the Cartesian product space limited by the maximum number of allowed class-k calls $N^k_{\max}$. Formally this state space is defined as

$$\tilde{\mathcal{I}} = \{0, \ldots, N^1_{\max}\} \times \cdots \times \{0, \ldots, N^K_{\max}\}.$$

In this space, the distribution (2.1) becomes a product of independent truncated Poisson distributions, and let $\mathbf{X}$ denote here temporarily a random variable having this distribution. Then, let us define K partitions of the space $\tilde{\mathcal{I}}$ into sets $\tilde{\mathcal{A}}^i_k$ where the sets in the k^{th} partition consists of k-columns in the space $\tilde{\mathcal{S}}$, as illustrated in Figure 2.4. Further, the conditional

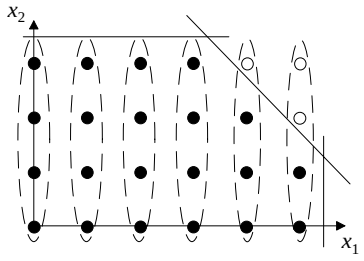

Figure 2.4: New partitioning of the state space.

expectations are defined

$$\begin{aligned} \eta_k(i) &= \mathrm{E}\left[1_{\mathbf{X}\in\mathcal{B}_k} \,|\, \mathbf{X} \in \tilde{\mathcal{A}}^i_k\right], \\ \vartheta_k(i) &= \mathrm{E}\left[1_{\mathbf{X}\in\mathcal{S}} \,|\, \mathbf{X} \in \tilde{\mathcal{A}}^i_k\right]. \end{aligned}$$

Note that the set $\mathcal{B}_k$ is still formed by the endpoints of the k-columns $\mathcal{A}^i_k$, not by those of the $\tilde{\mathcal{A}}^i_k$. Both of the conditional expectations can be calculated analytically,

$$\eta_k(i) = \begin{cases} 0, & \tilde{\mathcal{A}}^i_k \cap \mathcal{S} = \varnothing, \\ p_k(L^i_k)/g_k(N^k_{\max}), & \tilde{\mathcal{A}}^i_k \cap \mathcal{S} \neq \varnothing, \end{cases}$$

$$\vartheta_k(i) = \begin{cases} 0, & \tilde{\mathcal{A}}^i_k \cap \mathcal{S} = \varnothing, \\ g_k(L^i_k)/g_k(N^k_{\max}), & \tilde{\mathcal{A}}^i_k \cap \mathcal{S} \neq \varnothing, \end{cases}$$

where, again, L^i_k is the length of the column $\mathcal{A}^i_k$ ($\subseteq \tilde{\mathcal{A}}^i_k$), and $g_k(L) = \sum_{l=0}^{L} (\rho^l_k/l!)e^{-\rho_k}$.

With the aid of these conditional expectations the blocking probability (2.6) can be written as

$$B_k = \frac{\mathrm{E}[\eta_k(\iota_k(\mathbf{X}))]}{\mathrm{E}[\vartheta_k(\iota_k(\mathbf{X}))]},$$

where $\iota_k(\mathbf{X})$ now denotes the unique index i of set $\tilde{\mathcal{A}}_i^k$ to which $\mathbf{X}$ belongs. The corresponding Monte Carlo estimator becomes

$$\hat{B}_k = \frac{\sum_{n=1}^{N} \eta_k(\iota_k(\mathbf{X}_n))}{\sum_{n=1}^{N} \vartheta_k(\iota_k(\mathbf{X}_n))} = \frac{\sum_{n=1}^{N} p_k(L_k(\mathbf{X}_n)) 1_{\mathbf{X}_n^{(k)} \in \mathcal{S}}}{\sum_{n=1}^{N} g_k(L_k(\mathbf{X}_n)) 1_{\mathbf{X}_n^{(k)} \in \mathcal{S}}}, \tag{2.11}$$

where $L_k(\mathbf{X})$ denotes the length of the column $\mathcal{A}_k^i$ to which $\mathbf{X}$ belongs, $\mathbf{X}_n^{(k)}$ is the K-vector obtained from $\mathbf{X}_n$ by setting its k^{th} component to 0 (the set $\tilde{\mathcal{A}}_k^i$ to which $\mathbf{X}_n$ belongs has a nonempty intersection with $\mathcal{S}$ if and only if $\mathbf{X}_n^{(k)} \in \mathcal{S}$), and where we have utilized the fact that $g_k(N_{\max}^k)$ is constant for traffic class k and cancels out. Note that in this formulation even a sample $\mathbf{X}_n$ falling outside $\mathcal{S}$ can give a contribution to the numerator and denominator of (2.11). Additionally, (2.11) is a biased but consistent estimator, similarly as (2.7).

2.2.5 Variance Reduction by Importance Sampling

In what follows, the estimation of the blocking probabilities via the importance sampling simulation method is discussed. The main problem in the simulation is to quickly get a good estimate for β_k, i.e., the numerator in (2.3), especially in the case, when the probabilities B_k are very small. For completeness, it can be noted that the blocking probability B_k does not only depend on β_k, but also on the state sum $\gamma = \mathrm{P}\{\mathbf{X} \in \mathcal{S}\}$ given by the denominator of (2.3). However, this probability is usually close to 1 and it is easy to estimate by using the previously introduced conditional expectations method (or just the standard Monte Carlo method). Therefore, in the rest of this section we concentrate on efficient importance sampling based methods for estimating β_k.

Importance sampling is also a well known variance reduction method, see [32, chap. 4], and it is based on the following property. Consider the general problem of estimating the expectation $H = \mathrm{E}[1_{\mathbf{X} \in \mathcal{A}}]$, where $\mathbf{X} \in \mathcal{S}$ has distribution $p(\mathbf{x})$. Let $\mathbf{X}^*$ denote another random variable with distribution $P^* : p^*(\mathbf{x}) = \mathrm{P}\{\mathbf{X}^* = \mathbf{x}\} > 0$, $\forall \mathbf{X}^* \in \mathcal{A}$. With respect to this distribution the expectation H can be expressed as

$$H = \mathrm{E}[1_{\mathbf{X} \in \mathcal{A}}] = \mathrm{E}[1_{\mathbf{X}^* \in \mathcal{A}}\ w(\mathbf{X}^*)],$$

where $w(\mathbf{X}^*) = p(\mathbf{X}^*)/p^*(\mathbf{X}^*)$ is the so called likelihood ratio. This leads to the Monte Carlo estimator

$$\hat{H} = \frac{1}{N} \sum_{n=1}^{N} 1_{\mathbf{X}_n^* \in \mathcal{A}}\ w(\mathbf{X}_n^*). \tag{2.12}$$

The distribution $p^*(\cdot)$ is generally referred to as the importance sampling distribution, and the main difficulty in this method is to identify a distribution $p^*(\cdot)$ such that the variance of the estimator (2.12) is minimized. A well known result is that the ideal zero variance estimator for the problem is the original distribution $p(\mathbf{x})$ conditioned on the probability of the set $\mathcal{A}$, i.e., $p(\mathbf{X})/\mathrm{P}\{\mathbf{X} \in \mathcal{A}\}$. Unfortunately, this distribution is impractical, because, to compute the likelihood ratio for each sample requires explicit knowledge of the probability of

the generated sample. In the case of the ideal distribution, this would require knowledge of the very quantity being estimated. However, the point is to derive realizable IS distributions, which approximate the ideal distribution as closely as possible.

2.2.6 Composite Form IS Distribution and Exponential Twisting

Let $\mathbf{X}$ denote again the random variable defined in $\mathcal{I}$ with distribution (2.1), and consider estimating $\beta_k = \mathrm{E}[1_{\mathbf{X}\in\mathcal{B}_k}]$. Let $p^*(\mathbf{x})$ denote the IS distribution satisfying

$$P^* : p^*(\mathbf{x}) = \mathrm{P}\{\mathbf{X}^* = \mathbf{x}\} > 0 \quad \forall\ \mathbf{X}^* \in \mathcal{B}_k.$$

From (2.12) we have the following estimator

$$\widehat{\beta}_k = \frac{1}{N}\sum_{n=1}^{N} 1_{\mathbf{X}_n^*\in\mathcal{B}_k}\ w(\mathbf{X}_n^*), \tag{2.13}$$

where N is the number of generated samples of $\mathbf{X}^*$. In Publication 3 it is shown that the variance of the observed variable $1_{\mathbf{X}^*\in\mathcal{B}_k}\ w(\mathbf{X}^*)$ in (2.13) can be expressed as

$$\mathrm{V}\left[1_{\mathbf{X}^*\in\mathcal{B}_k}\ w(\mathbf{X}^*)\right] = \frac{\beta_k^2}{\beta_k^*} - \beta_k^2 + \beta_k^*\sigma^{*2}, \tag{2.14}$$

where β_k^* is the probability of the set $\mathcal{B}_k$ under P^*, $\beta_k^* = \mathrm{E}[1_{\mathbf{X}^*\in\mathcal{B}_k}]$, and σ^{*2} is the variance of the observed variable in the set of the blocking states under P^*, $\sigma^{*2} = \mathrm{V}[w(\mathbf{X}^*) \mid \mathbf{X}^* \in \mathcal{B}_k]$.

By increasing β_k^*, the probability of $\mathcal{B}_k$ under $p^*(\mathbf{x})$, the first and most important term in (2.14) is reduced. Ideally, if one can make $\beta^* = 1$, the first and second term completely cancel. Furthermore, if the probability of $\mathcal{B}_k$ can be increased uniformly, i.e., with $w(\mathbf{x})$ constant in $\mathcal{B}_k$, then $\sigma^{*2} = 0$. Thus the samples would have the distribution $\mathrm{P}\{\mathbf{X} = \mathbf{x} \mid \mathbf{X} \in \mathcal{B}_k\}$ and the estimator would have a zero variance. In practice one has to compromise between the following. It is important to increase the probability of the set $\mathcal{B}_k$ but at the same time it is important to keep $w(\mathbf{x})$ as constant as possible in $\mathcal{B}_k$ in order to minimize σ^{*2}.

Consider then restricting the whole set of possible IS distributions to the family of exponentially twisted distributions. In this case, $p(\mathbf{x})$ is a Poisson distribution with load $\boldsymbol{\rho}$. The exponentially twisted distribution is also a Poisson distribution, but with a different load. By using a single exponentially twisted distribution one can twist the mean of the sampling distribution to be, e.g., centered around the most probable blocking state (as is done in [25]). However, doing so can make the distribution of $w(\mathbf{x})$ in $\mathcal{B}_k$ uneven giving rise to a large σ^{*2} in (2.14). This may happen, because on the boundaries of all the link constraints, there can be states that contribute significantly to the state sum β_k, implying that an efficient IS distribution must be capable of producing samples lying on the boundaries of all the links that the traffic class uses. This can be achieved by using a composite distribution, which is

a weighted combination of several exponentially twisted distributions. This approach is supported by the results of Sadowsky and Bucklew in [33], which show that the asymptotically optimal twisting distribution for $\mathcal{B}_k$ indeed is of this composite form.

Now, the basic idea is to derive a distribution for traffic class k which will make all the blocking states associated with the active link capacity constraints more probable. To this end, one first needs to identify the most probable blocking states on the links which traffic class k uses in the network, i.e., on the links belonging in $\mathcal{R}_k$. Let $\mathbf{x}_j^*$ denote the most probable blocking state on link $j \in \mathcal{R}_k$. The most probable blocking state $\mathbf{x}_j^*$ on link j is found by maximizing (2.1) on the hyperplane representing the capacity constraint of the link,

$$\max_{\mathbf{x}} \prod_{k=1}^{K} \frac{\rho_k^{x_k}}{x_k!} e^{-\rho_k}$$
$$\text{subject to } \mathbf{x} \cdot \mathbf{b}_j = C_j,$$

where the components of $\mathbf{x}$ are real valued, i.e., not restricted to integers. This can be solved numerically, as shown in Publication 3. Then the composite sampling distribution for traffic class k is expressed as

$$p^*(\mathbf{x}) = \sum_{j \in \mathcal{R}_k} P_j \; p_j^*(\mathbf{x}), \tag{2.15}$$

where the P_j form a discrete probability distribution. In particular P_j, is the probability of using the distribution $p_j^*(\cdot)$ for generating the sample. Samples are generated from (2.15) by first drawing the index j of the link distribution $p_j^*(\mathbf{x})$ to be used from the discrete distribution of the P_j. The sample $\mathbf{X}_n^*$ is finally generated from the exponentially twisted distribution determined by the drawn link index and is used in the estimator (2.13).

This is illustrated in Figure 2.5, which depicts a two traffic class system with two link constraints. The left picture shows the equivalue contours of the original distribution $p(\mathbf{x})$, together with its maximum $\boldsymbol{\rho}$ and the points at which the equivalue surfaces touch the two link constraints, representing the most probable blocking states $\mathbf{x}_1^*$ and $\mathbf{x}_2^*$. The right picture shows the equivalue contours of the resulting composite distribution, which is a weighted combination of two twisted distributions having their maxima at the points $\mathbf{x}_1^*$ and $\mathbf{x}_2^*$. Note that this is not exactly what the method described above suggests. The method above suggested that the points $\mathbf{x}_1^*$ and $\mathbf{x}_2^*$ are used directly as the twisting parameters, which means that the expected value of the link sampling distributions $p_j^*(\mathbf{x})$ are equal to the $\mathbf{x}_j^*$. This does not imply that the maxima of the $p_j^*(\mathbf{x})$ are equal to the $\mathbf{x}_j^*$ – the maxima are close to the $\mathbf{x}_j^*$, but not exactly. However, for presentation purposes the picture has been obtained in the above mentioned manner, and the figure illustrates the idea behind the use of the composite distribution. The probabilities P_1 and P_2 in the composite distribution have here been obtained by simply using the relative weights of the points $\mathbf{x}_1^*$ and $\mathbf{x}_2^*$, i.e., $P_j = \mathbf{x}_j^*/(\mathbf{x}_1^* + \mathbf{x}_2^*)$ for $j = 1, 2$.

It remains to find a method to obtain the weights P_j. Again, the used heuristics are based on the goal of keeping the likelihood ratio as constant as possible in $\mathcal{B}_k$. Let J_k denote the number of links in the set $\mathcal{R}_k$. With J_k parameters P_j available, $w(\mathbf{x})$ cannot be made

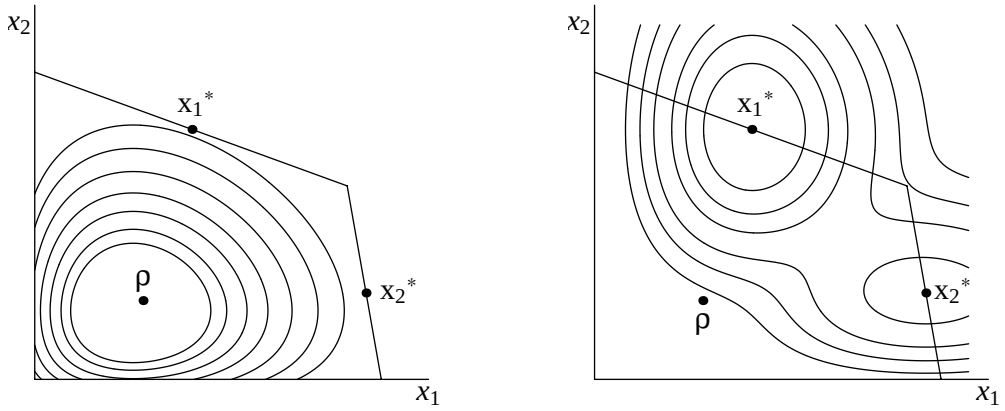

Figure 2.5: Equivalue contours of the original distribution $p(\mathbf{x})$ with load $\boldsymbol{\rho}$ (left figure) and the equivalue contours of the composite distribution $p^*(\mathbf{x})$ (right figure) for a two traffic class example.

constant everywhere in $\mathcal{B}_k$. Instead, one can choose to require $w(\mathbf{x})$ to be constant, α, at the most probable blocking states $\mathbf{x}_j^*$. This requirement leads to a set of linear equations

$$\sum_{i \in R_k} P_i \; p_i^*(\mathbf{x}_j^*) = \alpha \; p(\mathbf{x}_j^*), \quad \forall j \in R_k, \tag{2.16}$$

where the constant α is chosen such that $\sum_{j \in R_k} P_j = 1$. Unfortunately, there is no guarantee that the solution always satisfies $P_j \geq 0, \forall j \in R_k$. This approach is used in the numerical results given in Publications 3 and 4. If negative values appear, (2.16) may be replaced by a suitable minimization problem with the constraint, $P_j \geq 0, \forall j \in R_k$. Alternatively, one may simply apply the heuristics used in obtaining Figure 2.5, i.e., to use the relative weights of the points $\mathbf{x}_j^*$ as the P_j.

2.2.7 Decomposition and the Inverse Convolution IS Method

In the following, the most efficient importance sampling method, the inverse convolution method, developed during the course of the work and published in Publication 4, is presented. As discussed earlier, by using decomposition, it is possible to divide the task of estimating β_k into independent problems of estimating, for each $j \in \mathcal{R}_k$, expectations of the type

$$\xi_k^j = \mathrm{E}\left[\frac{1}{\nu_k^j(\mathbf{X})} \, 1_{\mathbf{X} \in \mathcal{S}} \, 1_{\mathbf{X} \in \mathcal{D}_k^j} \right],$$

in which case $\beta_k = \sum_j \xi_k^j$. Note that when estimating the ξ_k^j, the number of samples to be used for each estimate of ξ_k^j, N_j, are free parameters. The N_j can be chosen optimally by dividing the total number of samples in proportion to the standard deviations of the ξ_k^j. Of course, these standard deviations are not known before the simulation. Thus, a

dynamic sample allocation scheme is needed. A detailed description of the sample allocation algorithm is given in Publication 4.

To estimate each ξ_k^j efficiently importance sampling is applied. Obviously, in this case, the ideal IS distribution would always generate points that lie in $\mathcal{D}_k^j$, and are always inside the allowed state space $\mathcal{S}$, i.e., points that are in $\mathcal{B}_k$, with a distribution proportional to $p(\mathbf{x})/\nu(\mathbf{x})$. Consequently, the value of the observed variable $w(\cdot)/\nu(\cdot)$ would be a constant. This conditional distribution is unrealizable, but it can well be approximated by another conditional distribution

$$p_j^*(\mathbf{x}) = \mathrm{P}\{\mathbf{X} = \mathbf{x} \,|\, \mathbf{X} \in \mathcal{D}_k^j\} = \begin{cases} \dfrac{\mathrm{P}\{\mathbf{X} = \mathbf{x}\}}{\mathrm{P}\{\mathbf{X} \in \mathcal{D}_k^j\}} = \dfrac{p(\mathbf{x})}{v_k^j}, & \text{if } \mathbf{x} \in \mathcal{D}_k^j, \\ 0, & \text{otherwise,} \end{cases} \tag{2.17}$$

where v_k^j is the probability mass of the set $\mathcal{D}_k^j$. When using (2.17) as the IS distribution, the likelihood ratio is a constant,

$$w(\mathbf{x}) = \frac{p(\mathbf{x})}{p(\mathbf{x})/v_k^j} = v_k^j,$$

and the IS estimator for ξ_k^j becomes

$$\widehat{\xi}_k^j = \frac{v_k^j}{N_j} \sum_{n=1}^{N_j} \frac{1}{\nu(\mathbf{X}_n^*)} \, 1_{\mathbf{X}_n^* \in \mathcal{S}}, \tag{2.18}$$

where N_j denotes the number of samples.

This is illustrated in Figure 2.6. With IS distribution (2.17) points are generated in the set $\mathcal{D}_k^j$ (area between the dashed lines) and simulation is needed essentially only to determine which part of the probability mass of $\mathcal{D}_k^j$ is actually inside $\mathcal{S}$ (factor $1_{\mathbf{X}^* \in \mathcal{S}}$, grey area in the figure). Additionally, the factor $1/\nu_k^j(\mathbf{X}^*)$ compensates for double (or multiple) counting for such points $\mathbf{x}$ that belong to more than one of the sets $\mathcal{D}_k^j$. However, the effect of this is of minor importance as, in practice, most of the points belong to only one set $\mathcal{D}_k^j$.

The efficiency gain obtained with the above can be shown as follows (where, for clarity, the effect of the factor $1/\nu_k^j(\cdot)$ has been omitted). When using standard MC, the samples are generated from the original distribution $p(\cdot)$, and the task is to estimate the probability $p = \mathrm{E}[1_{\mathbf{X} \in \mathcal{S}} \, 1_{\mathbf{X} \in \mathcal{D}_k^j}]$. Each sample is then an independent Bernoulli variable and the relative error (or relative deviation) of the estimate, given by the ratio of the standard deviation and the expected value of the estimate, after N samples have drawn, is $\sqrt{(1-p)/(pN)}$. Thus, assuming for instance $p = 0.005$, almost 80 000 samples are needed to have a relative error of 5% for the estimate. On the other hand, when using (2.17) as the IS distribution, only the conditional probability $p' = \mathrm{E}\left[1_{\mathbf{X} \in \mathcal{S}} \,|\, \mathbf{X} \in \mathcal{D}_k^j\right]$ needs to be estimated. This probability is typically much greater than p. Thus, assuming for example that $p' = 0.9$ only about 45 samples are needed to reach the same 5% relative error level, giving a decrease by a factor of almost 2000 in the required sample size. Numerical experiments in Publications 4 and 5 verify that efficiency gains of this order or even greater can be obtained in practice, as well.

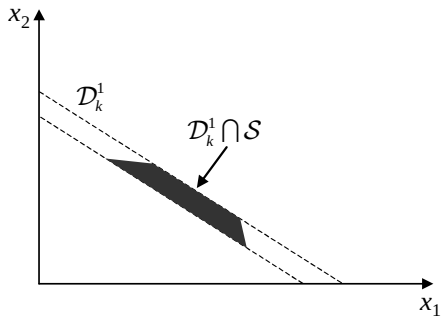

Figure 2.6: Estimation of ξ_k^j.

The Inverse Convolution Algorithm

One of the main contributions of this work is the so called inverse convolution method with which samples can be efficiently generated from the conditional distribution (2.17), as explained in Publication 4. As the estimation of ξ_k^j is considered for a fixed $j \in \mathcal{R}_k$, the link index j and the traffic class index k for which the β_k is to be estimated are omitted from the notation. This implies that C_j, b_k^j and $\mathcal{D}_k^j$ are denoted here by C, b and $\mathcal{D}$, respectively. To further simplify the notation, it is also assumed, without loss of generality, that the traffic classes which use link j have the indexes $1, \ldots, L$. The inverse convolution method is based on the observation that it is relatively easy to generate points into the set $\mathcal{D}$ from the conditional IS distribution (2.17), i.e., $\mathrm{P}\{\mathbf{X} = \mathbf{x} \,|\, \mathbf{X} \in \mathcal{D}\}$, by reversing the steps used to calculate the occupancy distribution of the considered link by convolutions.

Let Y_k^j denote the random variable for the occupancy of link j due to the traffic of class k, i.e $Y_k^j = b_k^j X_k$. The distribution of Y_k^j is

$$m_k^j(y) = \mathrm{P}\{Y_k^j = y\} = \begin{cases} f_k(x), & \exists\, x \in \mathbb{N} \,:\, y = b_k^j x, \\ 0, & \text{otherwise.} \end{cases}$$

Let S_l, with $l = 1, \ldots, L$, denote the occupancy on the considered link caused by the superposition of the first l classes, i.e.,

$$S_l = \sum_{l' \le l} Y_{l'}, \quad l = 1, \ldots, L.$$

We can also express $S_l = S_{l-1} + Y_l$, where S_{l-1} and Y_l are independent. The distribution of S_l, $q_l(x) = \mathrm{P}\{S_l = x\}$, can be obtained recursively from the convolution

$$q_l(x) = \sum_{y=0}^{x} q_{l-1}(x-y) m_l(y). \tag{2.19}$$

Thus, the probability mass of the set $\mathcal{D}$, v, is obtained from

$$v = \mathrm{P}\{\mathbf{X} \in \mathcal{D}\} = \mathrm{P}\{C - b < S_L \le C\} = \sum_{i=C-b+1}^{C} q_L(i).$$

The event $S_l = x$ is the union of the events $\{Y_l = y, S_{l-1} = x - y\}$, $y = 0, \ldots, x$ with the probabilities $m_l(y)q_{l-1}(x-y)$. Conversely, given $S_l = x$, the conditional probability of the event $Y_l = y$ is $m_l(y)q_{l-1}(x-y)/q_l(x)$, for $y = 0, 1, \ldots, x$. These probabilities can be precomputed and stored. Then, given $S_l = x$, using these probabilities it is easy to draw a value, say y, for Y_l and consequently for $S_{l-1} = x - y$. In fact, it is advantageous to store directly the values of the cdf

$$\mathrm{P}\{Y_l \leq y \,|\, S_l = x\} = \sum_{y'=0}^{y} m_l(y')q_{l-1}(x-y')/q_l(x). \tag{2.20}$$

Then the value of $Y_l \leq y$ can be drawn by finding the smallest y such that $\mathrm{P}\{Y_l \leq y \,|\, S_l = x\} \geq U$, where U is a random variable drawn from the uniform distribution in $(0, 1)$.

Now, S_L is the occupancy of the link, and the set $\mathcal{D}$ corresponds to $C - b < S_L \leq C$. A point in $\mathcal{D}$ can be generated by first drawing a value for S_L using the distribution $q_L(\cdot)$ conditioned on $C - b < S_L \leq C$, which is also precomputed and stored. Then, as described above, (Y_L, S_{L-1}) can be drawn from (2.20). Once the value of S_{L-1} is fixed, (Y_{L-1}, S_{L-2}) can be drawn. This process is continued until the value of the last component Y_1 has been drawn. The most important thing here is to note that the distributions of the conditional sets (Y_l, S_{l-1}) for a fixed value of S_l are easily precomputed and, hence, to generate each component Y_l, a uniformly distributed random variable is generated and the value of Y_l is obtained as an outcome from a simple table lookup. The other classes not using the link, i.e., classes $L + 1$ to K, are independent from classes $1, \ldots, L$ and from each other. Hence, their values are drawn independently from the distributions $f_k(\cdot)$, $k = L + 1, \ldots, K$. In summary the procedure for generating samples from (2.17) with the inverse convolution method can be described as follows. First we have the preparatory steps:

1. Compute the distribution of S_L recursively from (2.19).
2. Compute the conditional distributions given by (2.20), for $l = 1, \ldots, L$.

To generate a sample from (2.17), the following is performed:

1. Generate a value, say s, for link occupancy S_L from $\mathrm{P}\{S_L = x \,|\, C - b_L < S_L \leq C\}$.
2. For i from L downto 2
   - Generate the value of Y_i from the distribution $\mathrm{P}\{Y_i \leq y \,|\, S_i = s\}$ (eq. (2.20)).
   - Set $X_i \leftarrow Y_i/b_i$.
   - Set $s \leftarrow s - b_i X_i$.
3. Set $X_1 \leftarrow s/b_1$.
4. For i from $L + 1$ to K, generate X_i from $p_i(x) = (\rho_i^x/x!)\, e^{-\rho_i}$.

2.2.8 Decomposition and the Conditional Gaussian IS Method

In this section another IS distribution for estimating ξ_k^j is presented, which has been published in Publication 4. However, now the $p_j^*(\mathbf{x})$ will only approximately represent the conditional distribution (2.17), i.e., $\mathrm{P}\{\mathbf{X} = \mathbf{x} \,|\, \mathbf{X} \in \mathcal{D}_j^k\}$. On the other hand, the generation of samples from this distribution can be done without the precomputation and storage of a large number of probability tables as required in the inverse convolution method. Again, for ease of notation, the dependence on the traffic class k for which ξ_k^j is estimated and the link $j \in \mathcal{R}_k$ under consideration are omitted.

The idea of the method is briefly as follows (a full description is given in Publication 4). The distribution $\mathrm{P}\{\mathbf{X} = \mathbf{x} \,|\, \mathbf{X} \in \mathcal{D}\}$ can be approximated by a suitably parameterized conditional multinormal distribution. Sample points in $\mathcal{D}$ can be generated by first generating a value for the link occupancy from its marginal distribution in the strip $C - b < \mathbf{b} \cdot \mathbf{X} \leq C$ and then generating the other coordinates from a conditional multinormal distribution. As the normal distribution is a continuous distribution, the values have to be discretized by rounding them to the closest integers. There is, however, a small technical problem in the method. As will be explained, in order to be able to make the calculation of the likelihood ratio practical, the strip $C - b < \mathbf{b} \cdot \mathbf{X} \leq C$ needs to be enlarged somewhat, i.e., limits $r \leq \mathbf{b} \cdot \mathbf{X} \leq s$ are used, where $r < C - b + 1$ and $s > C$.

In more detail the procedure is as follows. First, the point $\mathbf{x}^*$ maximizing $p(\mathbf{x})$ on the constraining hyperplane $\mathbf{b} \cdot \mathbf{x} = C$ is determined (similarly as earlier when using the composite distribution and twisted distributions). Then, at that most important point $\mathbf{x}^*$ a Gaussian function $g(\mathbf{x})$,

$$g(\mathbf{x}) = \frac{a}{(\sqrt{2\pi})^K |\mathbf{\Gamma}|} \, e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m})^{\mathrm{T}} \mathbf{\Gamma}^{-1} (\mathbf{x}-\mathbf{m})},$$

is fitted to $p(\mathbf{x})$ (considered as a continuous function of $\mathbf{x}$). The parameters a, $\mathbf{m}$ and $\mathbf{\Gamma}$ of $g(\mathbf{x})$ are obtained by requiring the 0^{th}, 1^{st} and 2^{nd} derivatives of $p(\mathbf{x})$ to match those of $g(\mathbf{x})$ at $\mathbf{x}^*$. Since $p(\mathbf{x})$ is of the product form, the fitting reduces to k one-dimensional problems.

Let us further assume, without loss of generality, that the traffic class for which ξ is to be estimated is traffic class K. Next a change of variables is made where x_K is replaced by the occupancy of the link $\sum_k b_k x_k$, i.e.,

$$\begin{cases} z_i = x_i, \ \forall i = 1, \ldots, K-1, \\ z_K = \sum_{k=1}^{K} b_k x_k. \end{cases}.$$

In matrix notation, this and its inverse can be expressed as, with the components of $\mathbf{A}^{-1}$ obvious from the above equations,

$$\begin{cases} \mathbf{z} &= \mathbf{A}^{-1}\mathbf{x}, \\ \mathbf{x} &= \mathbf{A}\mathbf{z}. \end{cases}$$

Now it is easy to verify that if $\mathbf{X}$ is a random variable with distribution $N(\mathbf{m}, \mathbf{\Gamma})$ then $\mathbf{Z} = \mathbf{A}^{-1}\mathbf{X}$ is a random variable with distribution $N(\widetilde{\mathbf{m}}, \widetilde{\mathbf{\Gamma}})$, where

$$\begin{cases} \widetilde{\mathbf{m}} &= \mathbf{A}^{-1}\mathbf{m}, \\ \widetilde{\mathbf{\Gamma}}^{-1} &= \mathbf{A}^{\mathrm{T}}\mathbf{\Gamma}^{-1}\mathbf{A} \quad (\Rightarrow \widetilde{\mathbf{\Gamma}} = \mathbf{A}^{-1}\mathbf{\Gamma}(\mathbf{A}^{-1})^{\mathrm{T}}). \end{cases} \tag{2.21}$$

The conditional distribution of $\mathbf{X} \sim N(\mathbf{m}, \mathbf{\Gamma})$ conditioned on $r \leq \mathbf{X} \cdot \mathbf{b} \leq s$ corresponds to the conditional distribution of $\mathbf{Z} \sim N(\widetilde{\mathbf{m}}, \widetilde{\mathbf{\Gamma}})$ conditioned on $r \leq Z_K \leq s$. It is easy to generate $\mathbf{Z}$ from this distribution, and then $\mathbf{X}$ is obtained from $\mathbf{X} = \mathbf{A}\mathbf{Z}$.

To generate $\mathbf{Z}$, observe that Z_K obeys a univariate normal distribution $Z_K \sim N(\widetilde{m}_K, \widetilde{\Gamma}_{KK})$, and its value in the range $r \leq Z_K \leq s$ can be generated by any of the standard methods (e.g., by inversion of the cumulative distribution function, or, more efficiently, by the acceptance rejection method using an exponential majorizing function, see [32, chap. 2] or Publication 4, Appendix 2). Second, given the value of Z_K, the other components of $\mathbf{Z}$, i.e., $\mathbf{Z}^{(1)} = [Z_1, \ldots, Z_{K-1}]$ again obey a multinormal distribution by Theorem 10.2 in [20, p. 324]

$$\mathbf{Z}^{(1)} \sim N(\widetilde{\mathbf{m}}^{(1)} - \mathbf{B}_{11}^{-1}\mathbf{B}_{12}(Z_K - \widetilde{m}_K), \mathbf{B}_{11}^{-1}),$$

where $\widetilde{\mathbf{m}}^{(1)}$ denotes the first $K-1$ components of $\widetilde{\mathbf{m}}$ and the $\mathbf{B}_{ij}$, $i, j = 1, 2$, represent components in the partitioning of $\widetilde{\mathbf{\Gamma}}$

$$\widetilde{\mathbf{\Gamma}} = \begin{pmatrix} \overbrace{\mathbf{B}_{11}}^{K-1} & \overbrace{\mathbf{B}_{12}}^{1} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{pmatrix} \begin{matrix} \} K-1 \\ \} 1 \end{matrix} .$$

Note that $\mathbf{B}_{22} = \widetilde{\Gamma}_{KK}$. Thus,

$$\mathbf{Z}^{(1)} \sim \widetilde{\mathbf{m}}^{(1)} - \mathbf{B}_{11}^{-1}\mathbf{B}_{12}(Z_K - \widetilde{m}_K) + \mathbf{B}_{11}^{-1/2} \cdot \mathbf{N}^{(1)}, \tag{2.22}$$

where $\mathbf{N}^{(1)}$ is a vector of $K-1$ independent $N(0,1)$ distributed random variables. In summary, the procedure for generating samples into $\mathcal{D}$ can be described as follows. First we have the preparatory steps

1. Obtain $\mathbf{m}$ and $\mathbf{\Gamma}$ from the fitting procedure.
2. Calculate $\widetilde{\mathbf{m}}$ and $\widetilde{\mathbf{\Gamma}}$ from (2.21).
3. Determine the submatrices $\mathbf{B}_{ij}$ and calculate $\mathbf{B}_{11}^{-1}\mathbf{B}_{12}$ and $\mathbf{B}_{11}^{-1/2}$, needed in (2.22).

Then for each sample the following is performed:

1. Generate Z_K from $N(\widetilde{m}_K, \widetilde{\Gamma}_{KK})$ in the interval (r, s).
2. Generate $\mathbf{Z}^{(1)}$ using (2.22).
3. $\mathbf{X} = \mathbf{A}\mathbf{Z}$, where $\mathbf{Z} = [\mathbf{Z}^{(1)}, Z_K]$.
4. Round the components of $\mathbf{X}$ to closest integers.

Let $\mathbf{X}_n^*$ denote the samples obtained in the above described manner. In order to use the samples, it must be possible to calculate the likelihood ratio and therefore the probability of the generated samples, i.e., the integer lattice points in $\mathcal{D}$. Note that, because of the rounding operation, the probability of a sample $\mathbf{X}_n^*$ equals the probability mass of the conditional normal distribution within the K-dimensional unit cube with the center $\mathbf{X}_n^*$. If the cube is totally embedded in the strip defined by the condition $r \leq Z_K \leq s$, then the calculation is easy. Even though a change of variables is made in order to control the values of Z_K, the density in the strip $r \leq Z_K \leq s$ still is the product of Gaussian densities. Thus, the probability mass in the cube can be expressed as the product of the probabilities of the respective intervals of normal variables. On the other hand, if the cube is not wholly embedded in the strip $r \leq Z_K \leq s$, then the calculation of the probability is complicated.

Thus, we have to ensure that for each integer lattice point in $\mathcal{D}$ the surrounding unit cube is wholly embedded in the strip $r \leq Z_K \leq s$. This means that the strip must be enlarged by choosing $r = C - b + 1 - \Delta, s = C + \Delta$, where $\Delta = \frac{1}{2}\sum_k b_k$. This is illustrated in Figure 2.7, where the grey area corresponds to $\mathcal{D}$. Note that $\Delta = [\frac{1}{2}, \ldots, \frac{1}{2}] \cdot \mathbf{b}$, where $[\frac{1}{2}, \ldots, \frac{1}{2}]$ is the vector distance of the corner of the cube from the center. By enlarging the strip misses are inevitably generated from the set $\mathcal{D}$ and, to some extent, the performance of the sampling method is deteriorated.

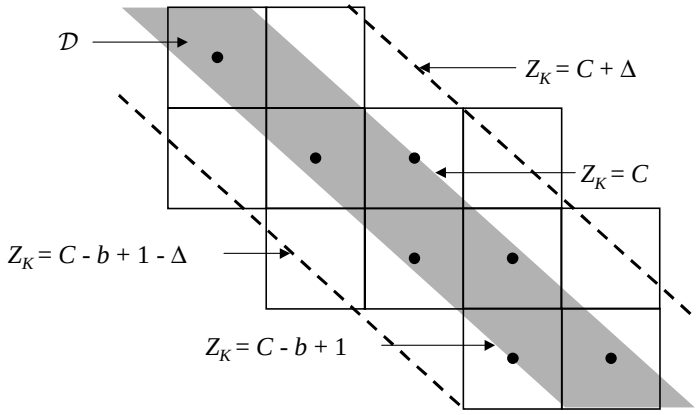

Figure 2.7: Enlargement of the generation interval of Z_K.

Remember that the Gaussian function $g(\cdot)$ is fitted such that it approximates the distribution $p(\cdot)$ as closely as possible at the point $\mathbf{x}^*$, i.e.,

$$p(\mathbf{x}) \approx g(\mathbf{x}) = a\, f_{N(\mathbf{m},\mathbf{\Gamma})}(\mathbf{x}),$$

where $f_{N(\mathbf{m},\mathbf{\Gamma})}$ denotes the probability density function of a normal distribution with mean $\mathbf{m}$ and covariance $\mathbf{\Gamma}$. Then, to be explicit, the IS distribution $p^*(\mathbf{x})$ approximating the conditional probability $\mathrm{P}\{\mathbf{X} = \mathbf{x} \,|\, \mathbf{X} \in \mathcal{D}\}$ is given by

$$p^*(\mathbf{x}) = \frac{\prod_{k=1}^{K} \left(Q(x'_k - \frac{1}{2}) - Q(x'_k + \frac{1}{2})\right)}{v}, \quad \mathbf{x} \in \mathcal{D},$$

where v is the total probability of the extended strip $v = Q(r') - Q(s')$, the primes refer to the normalized variables

$$x'_k = \frac{x_k - m_k}{\sigma_k}, \quad r' = \frac{r - \widetilde{m}_K}{\sqrt{\widetilde{\Gamma}_{KK}}}, \quad s' = \frac{s - \widetilde{m}_K}{\sqrt{\widetilde{\Gamma}_{KK}}},$$

and $Q(\cdot)$ denotes the tail probability function of the standard $N(0,1)$ distribution

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-y^2/2}\, dy = \frac{1}{2}\mathrm{erfc}(x/\sqrt{2}).$$

Finally, the estimator for ξ becomes

$$\begin{aligned} \widehat{\xi} &= \frac{1}{N}\sum_{n=1}^{N} \frac{1}{\nu(\mathbf{X}_n^*)}\, 1_{\mathbf{X}_n^* \in \mathcal{S}}\, 1_{\mathbf{X}_n^* \in \mathcal{D}}\, \frac{p(\mathbf{X}_n^*)}{p^*(\mathbf{X}_n^*)} \\ &= \frac{v}{N}\sum_{n=1}^{N} \frac{1}{\nu(\mathbf{X}_n^*)}\, 1_{\mathbf{X}_n^* \in \mathcal{S}}\, 1_{\mathbf{X}_n^* \in \mathcal{D}}\, \frac{p(\mathbf{X}_n^*)}{\prod_{k=1}^{K}\left(Q(X'_k - \frac{1}{2}) - Q(X'_k + \frac{1}{2})\right)}, \end{aligned}$$

where the $\mathbf{X}_n^*$ denote samples obtained with the above described procedure and generated into the enlarged strip.

2.2.9 Combining IS and Conditional Expectations

The use of exponentially twisted distributions is advantageous, because the generation of samples is then as simple as in standard MC (recall that an exponentially twisted Poisson distribution is also a Poisson distribution), and its use does not require a large amount of precomputed distributions either. The problem, on the other hand, is that the control over where the samples are generated is somewhat limited. However, it is possible to improve the "precision" of the sampling by combining the use of exponentially twisted distributions and the earlier defined method of conditional expectations.

Here the index of the traffic class k and the link index j are again reinstated in the notation. In [22] some early results were shown in this direction, where the IS heuristics of Ross in [31, chap. 6] were applied. The combination of IS and conditional expectations is, in this case, easy as the IS heuristics result in a single twisted IS distribution and the form of the conditional expectation function becomes such that one can again compute in advance all the required values of the function. To see this, consider the following. Assume that samples $\mathbf{X}_n^*$ are generated with an IS distribution $p^*(\mathbf{x})$ that is defined in the truncated state space $\tilde{\mathcal{I}}$ (i.e., the Cartesian product space limited by the maximum number of allowed class-k calls). When using the Ross heuristics $p^*(\mathbf{x})$ is also a Poisson distribution with a new load, say $\boldsymbol{\gamma}$. Also, assume that we have for traffic class k a partitioning of the state space $\tilde{\mathcal{I}}$ into subsets $\tilde{\mathcal{A}}_k^i$ and a partitioning of the state space $\mathcal{S}$ into subsets $\mathcal{A}_k^i$. Recall that each $\mathcal{A}_k^i$ is a subset of the corresponding $\tilde{\mathcal{A}}_k^i$, and that the set $\mathcal{B}_k$ is still formed by the endpoints of the

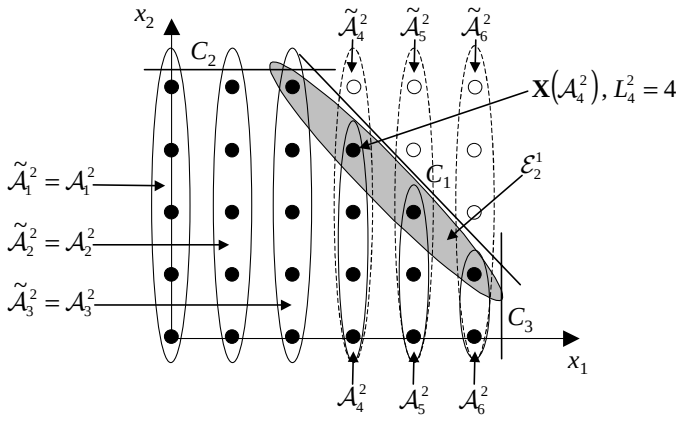

Figure 2.8: The sets $\tilde{\mathcal{A}}_k^i$ and $\mathcal{A}_k^i$ with examples of $\mathbf{X}(\mathcal{A}_k^i)$, L_k^i and $\mathcal{E}_k^j$ for a two traffic class example.

k-columns $\mathcal{A}_k^i$. Then the conditional expectation function, the value of which needs to be computed analytically, is

$$\eta_k(i) = \mathrm{E}\left[w(\mathbf{X}^*)\, 1_{\mathbf{X}^* \in \mathcal{B}_k} \,|\, \mathbf{X}^* \in \tilde{\mathcal{A}}_k^i\right]. \tag{2.23}$$

To compute this conditional expectation, let L_k^i denote the length of the k-column $\mathcal{A}_k^i$ (k-columns $\tilde{\mathcal{A}}_k^i$ have lengths $N_{\max}^k$) and $\mathbf{X}(\mathcal{A}_k^i)$ denotes the unique state that is the endpoint of the k-column $\mathcal{A}_k^i$. The sets $\tilde{\mathcal{A}}_k^i$ and $\mathcal{A}_k^i$, together with examples of $\mathbf{X}(\mathcal{A}_k^i)$ and L_k^i are depicted in Figure 2.8 for traffic class 2 in a two traffic class example. For future reference purposes, the picture also shows the set of states forming the set $\mathcal{E}_2^j$ for link 1. Additionally, before proceeding the function $q(x, \rho, N)$ is introduced, which gives the point probability of x for a Poisson distributed random variable with load ρ truncated at N, i.e.,

$$q(x, \rho, N) = \frac{\rho^x}{x!} \Bigg/ \sum_{n=0}^{N} \frac{\rho^n}{n!}\,.$$

Continuing with the evaluation of (2.23), one obtains $\eta_k(i) = 0$ if $\tilde{\mathcal{A}}_k^i \cap \mathcal{S} = \varnothing$ or else

$$\begin{aligned} \eta_k(i) &= w(\mathbf{X}(\mathcal{A}_k^i))\mathrm{E}\left[1_{\mathbf{X}^* \in \mathcal{B}_k} \,|\, \mathbf{X}^* \in \tilde{\mathcal{A}}_k^i\right], \\ &= w(\mathbf{X}(\mathcal{A}_k^i)) q(L_k^i, \gamma_k, N_{\max}^k) \end{aligned}$$

where we have utilized the fact that, in this case (thanks to the product form of $p^*(\mathbf{x})$), the conditional distribution of the states within each $\tilde{\mathcal{A}}_k^i$ is a simple one dimensional truncated Poisson distribution. Thus, to reduce the cost of evaluating the contribution of a given sample, it is also feasible to compute the different possible values of the conditional probability $\mathrm{E}\left[1_{\mathbf{X}^* \in \mathcal{B}_k} \,|\, \mathbf{X}^* \in \tilde{\mathcal{A}}_k^i\right]$ beforehand into tables.

However, the above described approach suffers from the properties of the IS heuristics, which uses a rather conservative twist. The combination of the composite distribution approach,

as discussed earlier, and the method of conditional expectations would also be possible, but the computation of the conditional expectation analytically is not anymore possible in advance. To be precise, the reason is the following. When $\mathbf{X}^*$ obeys the composite IS distribution (2.15) (again defined in $\tilde{\mathcal{I}}$), the IS distribution is not anymore of the product form. The conditional expectation function, in this case, equals $\eta_k(i) = 0$ if $\tilde{\mathcal{A}}_k^i \cap \mathcal{S} = \varnothing$ or else

$$\begin{aligned}\eta_k(i) &= w(\mathbf{X}(\mathcal{A}_k^i))\mathrm{E}\left[1_{\mathbf{X}^*\in\mathcal{B}_k} \,|\, \mathbf{X}^* \in \tilde{\mathcal{A}}_k^i\right] \\ &= w(\mathbf{X}(\mathcal{A}_k^i))\frac{p^*(\mathbf{X}(\mathcal{A}_k^i))}{\sum_{\mathbf{x}\in\tilde{\mathcal{A}}_k^i} p^*(\mathbf{x})} \\ &= w(\mathbf{X}(\mathcal{A}_k^i))\frac{\sum_{j\in\mathcal{R}_k} p_j^*(\mathbf{X}(\mathcal{A}_k^i))}{\sum_{\mathbf{x}\in\tilde{\mathcal{A}}_k^i}\sum_{j\in\mathcal{R}_k} p_j^*(\mathbf{x})},\end{aligned}$$

where each $p_j^*(\mathbf{x})$ is a product of Poisson distributions with different load parameters. From this one can observe that the conditional distribution of the points in the set $\tilde{\mathcal{A}}_k^i$ does not have a simple form depending only on the value of the k^{th} component of $\mathbf{X}$. Instead, it depends on the values of all the components of the given sample state making the precomputation of these conditional probabilities infeasible in practice. Note that it is still possible to compute the above conditional probability for a given generated sample state, but this requires that the conditional probability is computed on the fly making it somewhat impractical due to the amount of computations that need to be made.

In the above, the problem is really caused by the fact that the composite IS distribution (2.15) is not of the product form anymore. However, by using the decomposition result as discussed earlier, it is possible to decompose the problem of estimating β_k into separate problems of estimating the probabilities

$$\xi_k^j = \mathrm{P}\{\mathbf{X} \in \mathcal{E}_k^j\}.$$

The shape of each set $\mathcal{E}_k^j$ is much simpler than that of the set $\mathcal{B}_k$ and no composite distribution is required to sample the states in $\mathcal{E}_k^j$. Thus, to estimate each ξ_k^j a single twisted IS distribution can be applied and the method of conditional expectations can be used to further improve the sampling. To this end, $p_j^*(\mathbf{x})$ denotes here the same twisted link sampling distributions as in (2.15), i.e., $p_j^*(\mathbf{x})$ is a Poisson distribution defined in $\tilde{\mathcal{I}}$ with load $\boldsymbol{\gamma}_j$ being equal to the most probable blocking state on link j. The sets $\mathcal{A}_k^i$ and $\tilde{\mathcal{A}}_k^i$ are as earlier (see Figure 2.8). The conditional expectation function is

$$\eta_k(i) = \mathrm{E}\left[w(\mathbf{X}^*)\, 1_{\mathbf{X}^*\in\mathcal{E}_k^j} \,|\, \mathbf{X}^* \in \tilde{\mathcal{A}}_k^i\right].$$

Now, the IS distribution is, again, of the product form and the conditional distribution within a set $\tilde{\mathcal{A}}_k^i$ is, similarly as earlier, a one dimensional Poisson distribution. Thus, $\eta_k(i)$ equals

$$\eta_k(i) = \begin{cases} 0, & \mathcal{A}_k^i \cap \mathcal{E}_k^j = \varnothing, \\ w(\mathbf{X}(\mathcal{A}_k^i))\, q(L_k^i, \gamma_{jk}, N_{\max}^k), & \mathcal{A}_k^i \cap \mathcal{E}_k^j \neq \varnothing, \end{cases}$$

where γ_{jk} denotes the k^{th} component of $\boldsymbol{\gamma}_j$. The corresponding estimator for ξ_k^j can be expressed as

$$\hat{\xi}_k^j = \frac{1}{N} \sum_{n=1}^{N} 1_{\mathbf{X}(\mathcal{A}_k(\mathbf{X}_n^*)) \in \mathcal{E}_k^j} \, w\left(\mathbf{X}(\mathcal{A}_k(\mathbf{X}_n^*))\right) \, q(L_k(\mathbf{X}_n^*), \gamma_{jk}, N_{\max}^k)$$

where the notation $\mathbf{X}(\mathcal{A}_k(\mathbf{X}_n^*))$ has been used to denote the end point of the k-column $\mathcal{A}_k^i$ to which the state $\mathbf{X}_n^*$ belongs and $L_k(\mathbf{X}_n^*)$ denotes the length of the k-column in question. The estimator for β_k is then simply (as earlier) $\hat{\beta}_k = \sum_j \hat{\xi}_k^j$, and it is also possible to allocate the number of samples to be used for estimating each ξ_k^j optimally by the same allocation algorithm as described in Publication 4.

2.2.10 Comparisons between Methods

As mentioned earlier, the conditional expectations method is used in Publications 1–2. In contrast to the standard Monte Carlo method, where a sample gives a contribution to the estimator only when it hits $\mathcal{B}_k$, the use of conditional expectations allows the collection of the state information for an entire k-column to which a sample belongs. Furthermore, the method is simple and the values of all required functions can be computed analytically in advance. Thus, there is no extra overhead from using the method. Also, it can be used to estimate B_k for all k at the same time and it is independent of the method used to generate the samples. However, while being able to give a considerable variance reduction, even better results are obtained with the use of importance sampling.

The first steps in this direction were reported in [22], where the heuristics of Ross were used and, also, the combination of IS and conditional expectations. The use of the heuristics of Ross results in the same IS distribution parameters for all traffic classes, and, thus, the same samples can be used for estimating B_k for all traffic classes. Much better variance reduction results are obtained with the composite distribution approach in Publication 3 with exponentially twisted distributions fitted for sampling states around the most probable blocking state on each link that the considered traffic class uses. This method differs from the earlier conditional expectations methods in that the IS distributions are now separate for each traffic class (unless some traffic class happens to have the same route through the network). This means that to estimate the B_k for all K traffic classes one needs to perform K consecutive simulations (in the worst case), whereas with the earlier methods each generated sample can be used for the estimators of all B_k's. However, the variance reductions obtained with the composite approach amply compensate for this slight disadvantage.

Additionally, the method of conditional expectations can be combined with the use of exponentially twisted IS distributions to provide further variance reductions. As discussed earlier, this is easy in the case when the IS distribution consists of a single twisted distribution, but becomes impractical, if a composite of twisted IS distributions is used. Thus, the best variance reduction results when using exponentially twisted IS distributions would be obtained by applying the decomposition result, using a single exponentially twisted distribution combined with the method of conditional expectations to estimate each ξ_k^j and using

the optimal sample allocation algorithm to allocate the samples between each sub-problem. The advantage in using exponentially twisted distributions is that the sample generation is simple and fast, and requires only a small number of precomputations to be made. In this case also, when estimating B_k for all traffic classes, each class needs to be treated separately.

The greatest variance reductions are obtained by using the inverse convolution algorithm (described in Publication 4), which is based on using a suitable conditional distribution of the original distribution as the IS distribution. The fastest version of the algorithm requires a number of conditional distributions to be precomputed. However, then the generation of samples is as fast as in a standard MC method. Furthermore, the memory requirements of the algorithm, i.e., the number of elements in the arrays, are not prohibitive. The number of array elements to be stored can be seen to be $\frac{1}{2}KC(C+1)$ for each subproblem (there are at most J subproblems). It should be noted that the dependence on K is only linear whereas the size of the state space grows exponentially with K. However, if this memory requirement grows too large, the minimum requirement is that the distributions of the recursion (2.19), i.e., those of q_l and m_l, have been precomputed. In this case, the number of array elements to be stored is $2K(C+1)$ (K distributions of length $C+1$ for m_l's and q_l's, respectively) for each subproblem. Then the conditional distribution $\mathrm{P}\{Y_l \leq y \mid S_l = x\}$, given by (2.20), must be constructed on the fly, making the sample generation somewhat slower. The Gaussian approximation to the conditional distribution gives remarkably good results, as well. Although its performance is slightly degraded, as discussed earlier, due to the requirement of keeping the value of the likelihood ratio analytically computable. This caused the need to use slightly "looser" bounds for the sample generation, inevitably resulting in misses to be created from the target set. Still, the Gaussian method gives better results than the composite IS distribution approach. The Gaussian method does not require any precomputed distributions to be constructed, but the sample generation is not as efficient as in the inverse convolution method (or standard MC), where the sample is generated as an outcome of a table look-up and a random number. Finally, both the inverse convolution and the Gaussian method require also that each B_k is estimated separately.

2.3 Efficient Simulation of Multicast Loss Systems

In this section the application of the decomposition result and the use of the inverse convolution method are summarized in the case of the so called multicast loss system. For the full description see Publication 5.

2.3.1 The Multicast Loss System

Consider a network consisting of J links, indexed with $j = 1, \ldots, J$, link j having a capacity of C_j resource units. The set of all links is denoted by $\mathcal{J}$. The network is organized as a tree, where the root link is denoted by J. The set $\mathcal{U}$ denotes the set of user populations, located at the leaves of the tree. The leaf links and the user populations connected to them

are indexed with the same index $u \in \mathcal{U} = \{1, \ldots, U\}$. The set of links on the route from user u to the root node is denoted by $\mathcal{R}_u$. The user populations which use link j, i.e., for which link $j \in \mathcal{R}_u$, are denoted by $\mathcal{U}_j$. The size of the set $\mathcal{U}_j$ is denoted by U_j. The multicast network supports I channels, indexed with $i \in \mathcal{I} = \{1, \ldots, I\}$. The channels originating from the root node represent different multicast transmissions, from which the users may choose. Each channel has a capacity requirement $d_i \in \mathbf{d} = \{d_i; i \in \mathcal{I}\}$. Here the same capacity requirement is used for all links, but it is trivial to generalize this model to link dependent capacity requirements.

The state of a link j is defined by the states of the channels i. Each channel may be either *on* (1) or *off* (0), i.e., the state of channel i on link j is $Y_{j,i} \in \{0, 1\}$. The state of a link is denoted by the vector $\mathbf{Y}_j = \{Y_{j,i}; i \in \mathcal{I}\} \in \mathcal{S}$, where $\mathcal{S}$ is the link state space $\mathcal{S} = \{0, 1\}^I$. A multicast connection is defined by the pair (u, i) of the user u (leaf link) and channel i. Synonymously, the multicast connection (u, i) is also referred to as a traffic class. The network state $\mathbf{X}$ is defined by the states of the leaf links $\mathbf{Y}_u \in \mathcal{S}$,

$$\mathbf{X} = (\mathbf{Y}_u; u \in \mathcal{U}) = (Y_{u,i}; u \in \mathcal{U}, i \in \mathcal{I}) \in \Omega, \tag{2.24}$$

where $\Omega = \{0, 1\}^{U \times I}$ denotes the network state space. The state of link j is determined by the network state as follows:

$$\mathbf{Y}_j = \begin{cases} \mathbf{Y}_u & \text{if } j = u \in \mathcal{U}, \\ \bigoplus\limits_{u' \in \mathcal{U}_j} \mathbf{Y}_{u'} & \text{otherwise,} \end{cases} \tag{2.25}$$

where $\oplus$ denotes an or-operation between channel states in different links, i.e., channel i is *on* on link j if and only if there is a link $u \in \mathcal{U}_j$ on which the channel is *on*.

Calls (connections to the root of the tree) are generated at the leaf node u independently of other leaf nodes according to a Poisson process with intensity λ_u. Arriving calls select channel i with a probability α_i, the calls are always accepted if there is enough capacity on the links and the blocked calls are cleared. The holding times of channels can, again due to the insensitivity result, have a general distribution with a finite mean $1/\tau_i$. Thus the offered load for channel i at leaf node u is $\rho_{u,i} = \alpha_i \lambda_u / \tau_i$ and the *on*-probability for channel i at leaf node u, $p_{u,i}$ is

$$p_{u,i} = \mathrm{P}\{Y_{u,1} = 1\} = 1 - \exp(-\rho_{u,i}).$$

Note that this is not the only way of defining the call arrival process (see Nyberg et al. [27] for other variants).

In an infinite capacity system, all the traffic classes (u, i) are independent from each other. The steady state distribution of the leaf link u, $\pi_u(\cdot)$, is given by

$$\pi_u(\mathbf{y}) = \prod_{i \in \mathcal{I}} p_{u,i}^{y_i} (1 - p_{u,i})^{1 - y_i}, \tag{2.26}$$

The probability $p_{j,i}$ of channel i to be in the *on* state on any link j may be calculated from the corresponding known probabilities of the leaf links $\mathcal{U}_j$,

$$p_{j,i} = 1 - \prod_{u \in \mathcal{U}_j} (1 - p_{u,i}).$$

The steady state distribution $\pi(\mathbf{x})$ of the system is then

$$\pi(\mathbf{x}) = \mathrm{P}\{\mathbf{X} = \mathbf{x}\} = \prod_{u \in \mathcal{U}} \pi_u(\mathbf{y}_u).$$

The link capacity occupancy (or link occupancy for brevity) of link j is denoted by S_j,

$$S_j = \mathbf{d} \cdot \mathbf{Y}_j = \sum_{i=1}^{I} d_i Y_{j,i}.$$

In a finite capacity network, the state space is truncated by the capacity constraints of the links,

$$\tilde{\Omega} = \left\{ \mathbf{x} \in \Omega \,\middle|\, S_j \leq C_j, \forall j \in J \right\},$$

and the steady state distribution of the finite capacity system, $\tilde{\pi}(\cdot)$ is obtained by truncation

$$\tilde{\pi}(\mathbf{x}) = \mathrm{P}\{\mathbf{X} = \mathbf{x} \,|\, \mathbf{X} \in \tilde{\Omega}\} = \begin{cases} \dfrac{\pi(\mathbf{x})}{\mathrm{P}\{\mathbf{X} \in \tilde{\Omega}\}}, & \mathbf{x} \in \tilde{\Omega}, \\ 0, & \text{otherwise.} \end{cases}$$

In a finite capacity network, blocking occurs whenever a user tries to establish a connection for channel i, and there is at least one link $j \in \mathcal{R}_u$ where the channel is not already *on* and there is not enough spare capacity for setting the channel *on*. Without loss of generality, it is assumed that the channels are ordered so that the channel for which the blocking is calculated has the greatest index I. With this convention, the channel index is unnecessary and will be omitted for most of the notation. Let S'_j denote link occupancy due to the channels $\{1, \ldots, I-1\}$, i.e.,

$$S'_j = \sum_{i=1}^{I-1} d_i Y_{j,i}.$$

Then, the set of states where there is blocking for traffic class (u, I), $\mathcal{B}_u$ is defined as

$$\mathcal{B}_u = \left\{ \mathbf{x} \in \tilde{\Omega} \,\middle|\, \exists j \in \mathcal{R}_u : S'_j > C_j - d_I, \right\}.$$

The expression "link j blocks" is used if for link j, $S'_j > C_j - d_I$. Thus, the set $\mathcal{B}_u$ consists of the states where at least one link blocks. Then the time blocking probability for traffic class (u, I) is

$$B_u = \mathrm{P}\{\mathbf{X} \in \mathcal{B}_u \,|\, \mathbf{X} \in \tilde{\Omega}\} = \frac{\mathrm{P}\{\mathbf{X} \in \mathcal{B}_u\}}{\mathrm{P}\{\mathbf{X} \in \tilde{\Omega}\}}. \tag{2.27}$$

2.3.2 Decomposition and IS

The idea here is the same as earlier in the case of the multiservice loss system - the estimation of β_u,

$$\beta_u = \mathrm{P}\{\mathbf{X} \in \mathcal{B}_u\},$$

is divided to simpler sub-problems by partitioning $\mathcal{B}_u$ into sets $\mathcal{E}_u^j$. The set $\mathcal{E}_u^j$ is defined as the set of points in $\mathcal{B}_u$ where link j blocks but none of the links closer to user u block,

$$\mathcal{E}_u^j = \mathcal{B}_u \cap \Big\{ \mathbf{x} \in \Omega \,\Big|\, S'_j > C_j - d_I \wedge S'_{j'} \leq C_{j'} - d_I, \forall j' \in \mathcal{R}_u^j \Big\},$$

where $\mathcal{R}_u^j$ denotes the set of links on the path from u to j, including link u but not link j. With this definition the $\mathcal{E}_u^j$ form a partitioning of $\mathcal{B}_u$, similarly as was possible in the case of the multiservice loss system. Thus the estimation of $\mathrm{P}\{\mathbf{X} \in \mathcal{B}_u\}$ has been broken down to estimating the probabilities $\xi_u^j = \mathrm{P}\{\mathbf{X} \in \mathcal{E}_u^j\}$, $j \in \mathcal{R}_u$.

To estimate each $\mathrm{P}\{\mathbf{X} \in \mathcal{E}_u^j\}$, MC simulation with IS is applied. The IS distribution is of the same form as (2.17), i.e.,

$$p_j^*(\mathbf{x}) = \mathrm{P}\{\mathbf{X} = \mathbf{x} \,|\, \mathbf{X} \in \mathcal{D}_u^j\} = \begin{cases} \dfrac{\mathrm{P}\{\mathbf{X} = \mathbf{x}\}}{\mathrm{P}\{\mathbf{X} \in \mathcal{D}_u^j\}} = \dfrac{\pi(\mathbf{x})}{v_u^j}, & \mathbf{x} \in \mathcal{D}_u^j, \\ \\ 0, & \text{otherwise}, \end{cases} \tag{2.28}$$

where the set $\mathcal{D}_u^j$ corresponds to the set of blocking states for traffic class (u, I) in a network where only link j has a finite capacity C_j and all other links nave an infinite capacity,

$$\mathcal{D}_u^j = \Big\{ \mathbf{x} \in \Omega \,\Big|\, C_j - d_I < S'_j \leq C_j \wedge Y_{j,I} = 0 \Big\},$$

and v_u^j is the probability mass of the set $\mathcal{D}_u^j$, $v_u^j = \mathrm{P}\{\mathbf{X} \in \mathcal{D}_u^j\}$.

Let $\mathbf{X}_n^*$ denote a sample obeying (2.28). When using (2.28), the value of the likelihood ratio is a constant, $w(\mathbf{x}) = \pi(\mathbf{x})/p_j^*(\mathbf{x}) = v_u^j$, and the resulting IS estimator is

$$\begin{aligned} \widehat{\xi_u^j} &= \frac{1}{N_j} \sum_{n=1}^{N_j} 1_{\mathbf{X}_n^* \in \mathcal{E}_u^j} \, w(\mathbf{X}_n^*), \\ &= \frac{v_u^j}{N_j} \sum_{n=1}^{N_j} 1_{\mathbf{X}_n^* \in \mathcal{E}_u^j}. \end{aligned} \tag{2.29}$$

This estimation problem is equivalent to formulating the problem by means of expressing $\mathrm{P}\{\mathbf{X} \in \mathcal{E}_u^j\}$ as a conditional probability,

$$\mathrm{P}\{\mathbf{X} \in \mathcal{E}_u^j\} = \mathrm{P}\{\mathbf{X} \in \mathcal{E}_u^j \,|\, \mathbf{X} \in \mathcal{D}_u^j\} \mathrm{P}\{\mathbf{X} \in \mathcal{D}_u^j\},$$

from which $v_u^j = \mathrm{P}\{\mathbf{X} \in \mathcal{D}_u^j\}$ can be computed analytically, and estimator (2.29) is used to estimate the conditional probability $\mathrm{P}\{\mathbf{X} \in \mathcal{E}_u^j \,|\, \mathbf{X} \in \mathcal{D}_u^j\}$. The efficiency of this method compared with standard MC can be justified as earlier in the case of the multiservice loss system (see the explanations for Figure 2.6).

Finally, the estimator for β_u is simply

$$\hat{\beta}_u = \sum_{j \in \mathcal{R}_u} \hat{\xi}_u^j.$$

Given the total number of samples N to be used for the estimator, the number of samples N_j allocated to each sub-problem can be allocated optimally by in a similar fashion as discussed for the case of the multiservice loss system.

2.3.3 The Inverse Convolution Method

We are now only considering the estimation of one ξ_u^j for fixed $j \in \mathcal{R}_u$ and traffic class (u, I). Again, as in the case of the multiservice loss system, the following method is based on the observation that it is relatively easy to generate points from the conditional distribution $p_j^*(\mathbf{x}) = \mathrm{P}\{\mathbf{X} = \mathbf{x} \mid \mathbf{X} \in \mathcal{D}_u^j\}$ by reversing the steps used to calculate the occupancy distribution of the considered link. Note that the condition $\mathbf{X} \in \mathcal{D}_u^j$ is a condition expressed in terms of the occupancy, S_j', of the considered link. The idea in the inverse convolution method is to first generate a sample of $\mathbf{Y}_j$ such that the occupancy of the link is in the blocking region. Then, given the state $\mathbf{Y}_j$, the state of the network, i.e., states of the leaf links, is generated. The mapping $\bigoplus : \mathbf{X} \to \mathbf{Y}_j$ is surjective, having several possible network states $\mathbf{X}$ generating the link state $\mathbf{Y}_j$, and one of them is drawn according to their probabilities.

The main steps of the simulation can be summarized as follows:

1. Generate the states for leaf links u by
   (a) Generate a sample state $\mathbf{Y}_j$ under the condition $C_j - d_I < S_j' \leq C_j \wedge Y_{j,I} = 0$ for link j.
   (b) Generate the leaf link states $\mathbf{Y}_u$, $u \in \mathcal{U}_j$, with the condition that link j state $\mathbf{Y}_j = \oplus_{u \in \mathcal{U}_j} \mathbf{Y}_u$ is given.
   (c) Generate the states $\mathbf{Y}_u$, $u \in \mathcal{U} - \mathcal{U}_j$ for the rest of the leaf links as in the normal Monte Carlo simulation.
2. The sample state of the network $\mathbf{X}_n \in \mathcal{D}_u^j$ consists of the set of all sample states of leaf links generated with step 1.
3. To collect the statistics for estimator (2.29), check if $\mathbf{X}_n \in \mathcal{E}_u^j$.

Next, the steps 1a, and 1b above are explained in more detail.

Step 1a: Link State Sample Generation

First, the link occupancy S_j is easily calculated recursively as follows. Let $S_{j,i}$ denote link occupancy due to the first i channels,

$$S_{j,i} = \sum_{i' \leq i} d_{i'} Y_{j,i'}.$$

Then $S_j = S_{j,I}$ and $S_j' = S_{j,I-1}$. The $Y_{j,i}$ are mutually independent, and we can express $S_{j,i} = S_{j,i-1} + d_i Y_{j,i}$, where $S_{j,i-1}$ and $Y_{j,i}$ are independent. For sample generation, only the occupancy generated by the first $I - 1$ channels, S_j', is of interest, since a call for a channel cannot be blocked if it is already in the *on* state. This is also reflected in the definition of the set $\mathcal{D}_u^j$.

Let $\mathrm{P}\{S'_{j,i} = x\} = q_{j,i}(x)$ denote the probability distribution of $S'_{j,i}$. The $q_{j,i}(x)$ are obtained from

$$q_{j,i}(x) = q_{j,i-1}(x - d_i)p_{j,i} + q_{j,i-1}(x)(1 - p_{j,i}), \tag{2.30}$$

where the recursion starts with $q_{j,0}(x) = 1_{x=0}$. The probability mass v_u^j of the set $\mathcal{D}_u^j$, can be calculated as

$$v_u^j = \mathrm{P}\{\mathbf{X} \in \mathcal{D}_u^j\} = (1 - p_{j,I}) \sum_{i=C_j-d_i+1}^{C_j} q_{j,I-1}(i).$$

Note that in (2.30) the event $\{S_{j,i} = x\}$ is the union of the events $\{Y_{j,i} = y_{j,i}, S_{j,i-1} = x - d_i y_{j,i}\}$, $y_{j,i} \in \{0, 1\}$, where $y_{j,i} = 0$ means that the channel i is in the *off* state, and $y_{j,i} = 1$ means that the channel i is in the *on* state on link j. The corresponding probabilities are $q_{j,i'-1}(x)(1 - p_{j,i'})$ and $q_{j,i'-1}(x - d_{i'})p_{j,i'}$, respectively. Conversely, one can infer what is the conditional probability of the event $\{Y_{j,i} = 1,\, S_{j,i-1} = x - d_i\}$ given that $S_{j,i} = x$,

$$\mathrm{P}\{Y_{j,i} = 1,\, S_{j,i-1} = x - d_i \,|\, S_{j,i} = x\} = \frac{q_{j,i-1}(x - d_i)p_{j,i}}{q_{j,i}(x)}. \tag{2.31}$$

The probability of the event $\{Y_{j,i} = 0,\, S_{j,i-1} = x \,|\, S_{j,i} = x\}$ is then $1 - \mathrm{P}\{Y_{j,i} = 1,\, S_{j,i-1} = x - d_i \,|\, S_{j,i} = x\}$.

To generate a sample of the link state $\mathbf{Y}_j$ such that it belongs to $\mathcal{D}_u^j$, that is, having $C_j - d_I < S'_j \le C_j$ and $Y_{j,I} = 0$, we start by drawing a value for $S'_j = S_{j,I-1}$ using the distribution $q_{j,I-1}(\cdot)$ with the condition that $C_j - d_I < S'_j \le C_j$. This conditional distribution can be precomputed and stored. Given the value of $S_{j,I-1}$, the state $Y_{j,i}$ of each channel ($i = I-1, \ldots, 1$) is drawn in turn using the probabilities (2.31). Concurrently with the state $Y_{j,i}$, the value of $S_{j,i-1}$ becomes determined. This is then used as the conditioning value in the next step to draw the value of $Y_{j,i-1}$ (and of $S_{j,i-2}$), etc. Drawing a sample for $Y_{j,i}$ requires just generation of a uniform random number from the interval (0,1) and setting the channel to *on* state if the number is smaller or equal than the probability. Note that for reasonable sizes of links, it is advantageous to store the probabilities for fast generation of samples.

Step 1b: Generating Leaf Link States from a Given Link State

Having drawn a value for state $\mathbf{Y}_j$ of link j, it is possible to draw values of the state vectors $\mathbf{Y}_u$, $u \in \mathcal{U}$ of the leaf links. For $u \in \mathcal{U}_j$, states $\mathbf{Y}_u$ are generated under the condition $\mathbf{Y}_j = \oplus_{u \in \mathcal{U}_j} \mathbf{Y}_u$ using a similar inverse convolution procedure as above. This condition can be broken down to separate conditions for each channel, i.e., for each i we have a separate problem of generating the values $Y_{u,i}$, $u \in \mathcal{U}$, under the condition $Y_{j,i} = \oplus_{u \in \mathcal{U}_j} Y_{u,i}$ with a given $Y_{j,i}$. Note that only channels i which are in the *on* state on link j, $Y_{j,i} = 1$, need to be considered. If $Y_{j,i} = 0$, then necessarily $Y_{u,i} = 0$ for all $u \in \mathcal{U}_j$.

Consider first a convolutional approach for generating a link state for channel i and link j *if* the states for each link $u \in \mathcal{U}_j$ are already known. In this section, the index $u_j \in$

$\{1, \ldots, U_j\} = \mathcal{U}_j$ is used for the subset of leaf links. Let $S_{u_j,i} = x$ denote the event that the channel is in state x on link j when $u' = 1, \ldots, u_j$ leaf links have been counted for, i.e.,

$$S_{u_j,i} = \bigoplus_{u' \le u_j} y_{u',i}.$$

Let $\mathrm{P}\{S_{u_j,i} = 1\} = q_{u_j,i}$ be the probability that the channel is on in at least one of the links $\{1, \ldots, u_j\} \subset \mathcal{U}_j$. These probabilities can be calculated recursively from

$$q_{u_j,i} = 1 - (1 - p_{u_j,i})(1 - q_{u_j-1,i}) = (1 - q_{u_j-1,i})p_{u_j,i} + q_{u_j-1,i}. \tag{2.32}$$

The recursion starts with $q_{0,i} = 0$. If $S_{u_j-1,i} = 1$, then necessarily $S_{u_j,i} = 1$. Conversely, to generate the state for each leaf link, given the value of $Y_{j,i}$, it is first checked if the channel would be *off* after $u_j - 1$ leaf links counted for, since if it is, then it has to be *on* on leaf link u_j, and *off* on the rest of the links, which happens with probability

$$\mathrm{P}\{S_{u_j-1,i} = 0 \,|\, S_{u_j,i} = 1\} = \frac{(1 - q_{u_j-1,i})p_{u_j,i}}{(1 - q_{u_j-1,i})p_{u_j,i} + q_{u_j-1,i}} = \frac{(1 - q_{u_j-1,i})p_{u_j,i}}{q_{u_j,i}}. \tag{2.33}$$

Note that the event $S_{u_j-1,i} = 0$ implies directly that $\{Y_{u',i} = 0, \forall u' < u_j\}$. The probability $\mathrm{P}\{S_{u_j-1,i} = 1 \,|\, S_{u_j,i} = 1\}$ is given by $1 - \mathrm{P}\{S_{u_j-1,i} = 0 \,|\, S_{u_j,i} = 1\}$. In the case of the event $S_{u_j-1,i} = 1$, one needs to check if the channel is *on* on the leaf link u, which happens with probability

$$\mathrm{P}\{Y_{u_j,i} = 1 \,|\, S_{u_j-1,i} = 1\} = p_{u_j,i}.$$

Thus, a uniform random number is generated first to decide if the leaf link is the last one for which the channel is on. If it is not, yet another uniform random number is generated to decide if the channel should be set on. This procedure is repeated for each channel. As a result of this procedure, the state vectors of each leaf link $u \in \mathcal{U}_j$ are obtained. The rest of the leaf link states must be generated as in the normal Monte Carlo simulation using eq. (2.26).

2.4 Conclusions

The work in this chapter presents a series of methods with which the estimation of the blocking probabilities in multiservice loss systems can be made quicker and/or more accurate. Additionally, in Section 2.3 the application of the decomposition principle and the inverse convolution algorithm in multicast networks is presented. As mentioned earlier, the best results are obtained by using IS (the composite distribution approach) and combining IS with the decomposition principle (the inverse convolution method, and its Gaussian approximation). Of these, the use of the decomposition result and the inverse convolution method together with the optimal sample allocation algorithm, as described in this thesis and Publication 4, represents the most efficient way of performing the simulation. Qualitative reasons for the efficiency of this approach can be listed as follows. The decomposition

allows us to break the problem into more easily manageable subproblems. For each subproblem, the inverse convolution method produces samples from a nearly optimal IS distribution, generating typically only few misses. The dynamic optimal sample allocation between different subproblems eliminates the need to know in advance the relative importance of each subproblem, which is a problem for the composite distribution when fixing the composite weights.

Numerical results on the relative efficiency of the methods presented in this thesis compared with results obtained by using standard MC or known heuristics (namely, the heuristics of Ross and Mandjes) are presented in Publication 4, Table 1, which gives the relative deviation (ratio of the standard deviation to the mean) of the estimates of $\hat{\beta}_k$ for various cases. However, in Table 2.1, the results have been converted to give directly the gain in terms of the required sample size using standard MC as a reference. In Table 2.1, Cases 1 and 2 correspond to a small network example with B_k of the order of 10^{-2} and 10^{-4}, respectively, and Cases 3 and 4 to a large network example with B_k of the order of 10^{-3}. For these cases, no results are available corresponding to the use of the conditional expectation method or its combination with IS. However, as illustrated by the numerical results in Publication 1, the use of the conditional expectations method gives reductions slightly better than those obtained by using the Ross heuristics. The combination of IS and the conditional expectation method, as discussed in Section 2.2.9, has not been tested in practice, but presumably its performance is comparable to that of the Gaussian method. As can be seen from Table 2.1, the inverse convolution method gives outstanding efficiency gains even if the blocking probabilities are in the "normal" range $10^{-4} \ldots 10^{-2}$. If the estimated blocking probabilities were smaller, even greater efficiency gains would be observed.

Table 2.1: The efficiency gain of the different methods in terms of the sample sizes using the standard MC as a reference.

Case	Convolution	Gaussian	Composite	Single twist	Ross
1	1853	50	4	3	2
2	$1.01 \cdot 10^6$	5700	316	125	16
3	18 418	90	9	9	2
4	669	38	5	3	2

Regarding the generality of the methods described in this thesis, they can be basically used in loss systems for estimating any performance measure expressed in terms of state sums consisting of states related to the occupancy levels of the links. On the other hand, all the methods presented in this thesis rely heavily on the structural properties of loss systems. Of these, the most important property is the product form of the steady state distribution. The second property is that the capacity limitations of the system are linear constraints on the state space expressed in terms of a single variable, the occupancy of a given link. These are properties that follow directly from the assumptions of the system, i.e., Poisson arrivals, the bandwidth requirement of each class has a fixed value, the effective bandwidth, on each link it uses, and the assumption of fixed routing (i.e., there is only one possible route for the

connection through the network). Of these, perhaps the most restrictive assumption from the point of view of generality is the assumption of fixed routing. Thus, for instance, it is not possible to use these simulation methods for comparing the performance of different routing mechanisms. However, given that the loss system model applies, it is very hard to imagine any new methods for estimating the blocking probabilities giving better results, without significantly increasing the computational burden, than the inverse convolution method combined with the dynamic optimal sample allocation scheme. Thus, future research efforts could be directed towards finding suitable problem areas, where loss systems or other systems possessing sufficiently similar properties can be used for modeling, similarly as is done in this thesis for the multicast network.

Chapter 3

Analysis of the RED Buffer Management Algorithm

3.1 Introduction

Traffic volumes in the Internet are growing at a rapidly increasing rate. This growth in traffic creates great challenges for controlling and managing the traffic flows in the network. In such a situation, one of the most important issues is congestion control, i.e., how the network should react in situations where the traffic load becomes too high, threatening the stable operation of the network.

In the Internet, this problem has been traditionally handled by the use of the Transmission Control Protocol (TCP), which is a reliable packet transfer protocol aiming at providing each TCP flow a fair share of the limited bandwidth. In the current Internet a vast majority of the data (e.g., HTML, FTP, e-mail, and telnet traffic) is transmitted by using TCP. However, it has been discovered that the use of TCP is not enough to guarantee the stable operation of the network and that TCP alone does not always achieve a fair sharing of the bandwidth. In addition, the amount of traffic generated by non-responsive sources, e.g., UDP sources, has been steadily rising with the increased use of the Internet for transmitting real time traffic streams, such as real-time audio and video. In general, non-responsive flows do not react to congestion by cutting down their sending rates and can thus obtain a more than their fair share of the bandwidth at the expense of the responsive TCP flows.

For these reasons, the Internet Engineering Task Force (IETF) has recommended in their standards the use of new congestion control mechanisms, so called Active Queue Management (AQM) methods that are implemented in the buffers of the routers of the network, see [42] (RFC2309). Basically, these mechanisms attempt to reduce the unfairness between responsive and unresponsive flows, and to inhibit the build up of congestion in the buffers by implicitly signaling the responsive traffic sources to reduce their sending rates already before the buffer becomes overloaded.

One of the most prominent AQM methods is the Random Early Detection (RED) algorithm proposed by Floyd and Jacobson in [50]. The algorithm has also been implemented in commercially available routers. It has been designed to work in cooperation with the TCP sources (without considering the problem of unresponsive sources), and in [50] it has been shown by using simulations that the algorithm is able to alleviate the problem of global synchronization of the TCP sources, which may happen when a congested buffer overflows, as well as to increase fairness between high speed and low speed TCP flows.

In this part of the thesis the aim is to create an analytical model of the interaction between a large TCP population and a RED controlled queue such that the impact of the different system parameters on the performance can be explored. The results have been published in Publications 6–8. The model for the RED controlled queue is defined in Publication 6 based on earlier work by Sharma et al. in [71] to account for the oscillations observed in the simulations of [71]. In Publication 7, a dynamic model is derived for the interacting TCP-RED system. In Publication 8, the stability analysis of the model is performed, where, not only sufficient, but also necessary conditions are derived for the system to be asymptotically stable.

3.1.1 Review of TCP-RED Analysis

A considerable amount of research has been devoted to the performance analysis of the RED algorithm. Primarily the methods used have been simulation based and have been aimed at solving some of the deficiencies of the original RED algorithm. Feng et al. [47] propose an adaptive discarding mechanism where the maximum discarding probability parameter of RED is varied according to the number of flows present in the system. Lin and Morris [62] showed that RED is not efficient in the presence of non-adaptive (or "non-TCP-friendly") flows, such as UDP flows, and for that they propose a per flow version of RED (FRED). The problem of isolating misbehaving and well-behaved flows and providing specialized treatment for such circumstances has resulted, besides FRED, in a number of algorithms: RED+ [76], SRED [67], BRED [40], REM [41] and stochastic fair BLUE [48]. Clark and Fang [45] have suggested the use of another variant of RED, called RIO, to provide different classes of services in the Differentiated Services framework.

In addition to modifications just in the packet discarding algorithm, there exists a vast literature dealing with defining alternatives for the TCP flow control. The purpose is then to derive new flow control algorithms such that the network resources are shared in a "fair" manner. One popular framework has been the so called congestion pricing approach (suggested by Kelly in [58]). In this approach, a user's perceived utility from receiving a given amount of bandwidth is measured by a utility function. The network provides users with information about the current congestion state of the network in the form of pricing information and the users are free to react to these signals as they please. It turns out that if the users react "sensibly", this defines a flow control algorithm that also maximizes the system's utility (the sum of the utilities of all users). In this thesis, however, the focus is not on the different RED variants that exist nor on alternative congestion control strategies.

Thus, the references mentioned above represent only chosen examples from the literature.

The steady state modeling of TCP under random packet losses has also received a lot of attention among the research community. Simple models have been developed, e.g., in Mathis et al. [63], in Padhye et al. [68], and in Floyd [52], where the well known result has been derived that the steady state throughput of a TCP connection is proportional to $1/(R\sqrt{p})$, with R denoting the round trip time and p the packet loss probability. More recent and accurate models (e.g., in terms of the assumptions regarding the packet loss process) providing results for the distribution of the TCP congestion window in steady state can be found, for example, in Misra and Ott [65], in Brown [43], and in Altman et al. [38, 39].

Recently, analytical approaches have been also used for the performance analysis of RED controlled buffers. May et al. [64] have developed a simplified model of the RED algorithm. However, RED is more complicated and several of its important features (e.g., the average queue length process) are not captured in this model. A more detailed analysis has been presented by Peeters and Blondia in [69]. Sharma et al. [71] have analyzed the RED algorithm in complete detail and have also developed asymptotic approximations for easing the burden of computing the performance indices of interest. By using a similar approach as in [71], the performance of a RIO controlled queue has been studied by Kuusela and Virtamo in [60]. Köhler et al. [61] present a discrete time Markovian model for the TCP-RED interaction. A fixed point model for a network of AQM routers is given by Bu and Towsley in [44].

In the work presented here the aim has been to develop a dynamic model for the interaction between an idealized TCP population and a RED controlled queue. The work builds on one of the ideas presented in [71], namely the quasi-stationary approximation for modeling the time dependent behavior of the system in the "mean" sense by differential equations. Related work has been done by Firoiu and Borden in [49], where a system of difference equations is derived and some qualitative analysis is given regarding the stability. From the point of view of the present work, the independent modeling work by Misra et al. in [66] is similar to the approach used in this thesis. However, the modeling technique in [66] differs from the present one in some respects, and the work in [66] can be considered complementary to the one described here. In the present work, a more detailed model for the queue dynamics is developed (in Publications 6 and 7), while [66] focuses more on the source dynamics. In the RED buffer model, [66] assumes an infinite and non-emptying buffer and the model derivation results in an additional modeling parameter to be tuned. On the other hand, the queue model described here takes overflows and empty queues into account, and does not have additional parameters. Also, the term describing the increase in the TCP window size is more accurate in the present model, since the present model takes into account the fact that the rate of acknowledgments back from the receiver depends on the sending rate at the time when the packets were sent. Additionally, some of the features (e.g., queue length dependent RTTs and the network case) of the model in [66] can be incorporated into the model described here, as well. The stability analysis of the model from [66] is given by Hollot et al. in [56]. The authors make a simplified linearization of the original system and are able to give necessary conditions for the system to be asymptotically stable. On the

other hand, Publication 8 gives, not only necessary, but sufficient conditions for the system derived in Publication 7 to be asymptotically stable without making any simplifications in the linearization, albeit an approximation is made in the model of Publication 7 to reduce its dimensionality. Recently, in [57] the authors of [56] have also presented an alternative AQM mechanism to replace RED, which has a better convergence speed than RED, by using a so called PI controller instead of a low pass filter, which the RED algorithm essentially is.

3.1.2 Overview of the Chapter

Short overviews of the TCP congestion control and the RED algorithm are given in Sections 3.2 and 3.3, respectively. The main results of Publications 6–8 have been summarized in Sections 3.4 and 3.5. Section 3.4 presents the derivation of the dynamic analytic model for the TCP-RED interaction, and in Section 3.5 the results on the stability analysis of the system are reviewed.

3.2 TCP Overview

Originally TCP did not implement any congestion control algorithms. The flow control was simply based on a maximum allowed window size (i.e., a maximum amount of unacknowledged data in transit) as determined by the receiver side and the transmission policy that new packets could only be sent when an acknowledgement (ACK) for a sent packet was received, signaling that a packet has been successfully received and has left the network. The use of the flow of ACKs to pace the transmission of new packets has been generally referred to as the self clocking property of TCP.

TCP congestion control was introduced to the Internet in the late 80's after the occurrence of so called congestion collapse(s), which can occur when sources are able to send traffic uncontrollably according to the Go-Back-N retransmission scheme of the original TCP. During a time of congestion the network is then transmitting mostly retransmitted packets, which have been lost due to congestion (buffer overflow), resulting in degraded throughput of the network. For this reason, TCP Tahoe was released in 1988 including three new algorithms for congestion control: slow start, congestion avoidance and fast retransmit. TCP Reno was released in 1990 and it added one more algorithm to the previous flow control algorithms called fast recovery. Next these four algorithms are briefly explained. More detailed descriptions of these algorithms can be found, e.g., in [72] (RFC2001) and [70, chap. 6.3]. TCP Tahoe is discussed first (slow start, congestion avoidance, fast retransmit) and then the effect of the addition of fast recovery in TCP Reno is explained.

In TCP, the receiver can regulate the data transmission rate by using its advertised window size, $awnd$, sent in each ACK. Also, before a TCP connection opens (i.e., any data packets are sent), the receiver advertises the packet size, MSS (maximum segment size), to be used during the transmission. The TCP congestion control introduced two new state variables for the connection: the congestion window, $cwnd$, and the slow start threshold, $ssthresh$.

Additionally, the definition of the window size of the sender, w (i.e., the maximum amount of unacknowledged data that can be in transit), was changed to

$$w = \min(cwnd, awnd),$$

instead of being equal to *awnd* as it used to be. This reflects the fact that the TCP flow control can be either network bandwidth limited (*cwnd*) or receiver capacity limited (*awnd*). The role of the TCP congestion control is, through adjusting *cwnd*, to try to reach a steady state as quickly as possible reflecting the available bandwidth in the network, and to react to the changing load conditions during the lifetime of the connection.

The TCP congestion control uses lost packets as control information and modifies the value of *cwnd* upon detecting packet losses. Packet losses can be detected in two ways: either by a time out mechanism or by so called duplicate ACKs. Duplicate ACKs refer to a situation where the receiver notices a missing packet but is still receiving a stream of packets. The receiver then transmits an ACK for each received packet, but the sequence number of the acknowledged packet is not increased due to the missing packet. Thus, the sender observes a number of ACKs that are acknowledging the same packet and determines that the packet following the packet, for which the duplicate ACKs were received, has been lost.

In the following discussion it is assumed that the limitation is always caused by the value of *cwnd*, i.e., that $w = cwnd$. Also, note that in the real TCP, data transmission occurs in bytes, but the explanations here are given as if the transmission is based on discrete packets of size MSS bytes. Thus, in the text here, a window of size w corresponds to $(w \cdot MSS)$ bytes of data. Also, upon receiving a non-duplicate ACK, a connection is allowed to transmit an amount of bytes equaling the difference between the current window size w and the amount of unacknowledged bytes that are in transit (which is given, in turn, by the difference between the sequence number of the last sent byte and the last acknowledged byte).

Slow Start and Congestion Avoidance (TCP Tahoe)

In the original TCP, when a TCP connection started, the source transmitted immediately a full window of data, i.e., *awnd* packets, and started to wait for ACKs, resulting in the flooding of the buffers of low speed links along the route of the TCP connection. To avoid this the slow start algorithm was introduced.

When a new TCP connection is established, the connection starts with slow start and sets $cwnd = 1$ (and *ssthresh* is given some default value). Then, for each received ACK, *cwnd* is updated according to

$$cwnd \leftarrow cwnd + 1,$$

implying roughly a doubling of *cwnd* (and the sending rate) for each round trip time (exponential growth). The exponential growth continues until a packet loss is observed, either via a time-out or duplicate ACKs. At that point, *ssthresh* is updated to contain the value of half of the window size just before the loss, i.e.,

$$ssthresh \leftarrow cwnd/2,$$

and the connection falls back to slow start. In a sense, the threshold *ssthresh* now contains a rough estimate of the available bandwidth for the connection.

As mentioned, after the first packet loss, the connection starts from slow start, with $cwnd = 1$, and starts increasing its window exponentially until the $ssthresh$ is reached. At this point the connection switches from slow start mode to congestion avoidance. The idea is that because the connection has now reached the previous estimate of the available bandwidth for the connection ($ssthresh$), the connection should not anymore try to increase its sending rate aggressively, but only moderately to take advantage of the free bandwidth that has become available since the previous measurement of the available bandwidth. Thus, after reaching $ssthresh$ the congestion window is updated for each ACK as

$$cwnd \leftarrow cwnd + 1/cwnd,$$

implying an increase by 1 for $cwnd$ successfully transmitted packets, i.e., a linear increase in time. This is continued until a loss is again detected either by time out or duplicate ACKs, in which case the connection falls back to slow start.

Fast Retransmit (TCP Tahoe)

The idea in fast retransmit is to speed up the process of retransmissions in TCP. It is based on the observation that when a source receives duplicate ACKs, they can have two possible causes: either a packet is missing or the packets have gotten out of order in the network (for some external reason). The source cannot know which is the case immediately from the first duplicate ACK. However, if several duplicate ACKs are received, it is reasonable to assume that the packet has been lost. Then, instead of waiting for a time out to happen (which occurs at a relatively slow time scale compared to ACK arrivals when dealing with large windows), the source waits typically for three duplicate ACKs and immediately retransmits the (presumably) missing packet and goes back to slow start.

Fast Recovery (TCP Reno)

As can be noticed from the above, in TCP Tahoe the connection falls back to slow start every time there is a packet loss (detected either by a time out or duplicate ACKs). However, when the transmission windows are large and packet losses relatively infrequent such that the connection rarely loses many packets in a row, it is better to have the connection stay in congestion avoidance and not fall back to slow start after packet loss, since when windows are large it takes some time for the window to grow from 1 to $ssthresh$.

The idea in fast recovery is exactly to enforce this behavior. This is only done in connection with fast retransmit, because then, when windows are large and assuming that only, e.g., 1 packet has been lost, the TCP source is still receiving a flow of duplicate ACKs corresponding to packets that have arrived and have been stored at the receiver. The TCP source can then use these ACKs to maintain the self clocking of the packet transmissions. Thus, when fast retransmit performs the retransmission of the (supposedly) missing packet after 3 duplicate ACKs, the connection does not fall back to slow start, but, after a short transition, the congestion window has been set to

$$ssthresh \leftarrow cwnd/2,$$

and

$$cwnd \leftarrow ssthresh.$$

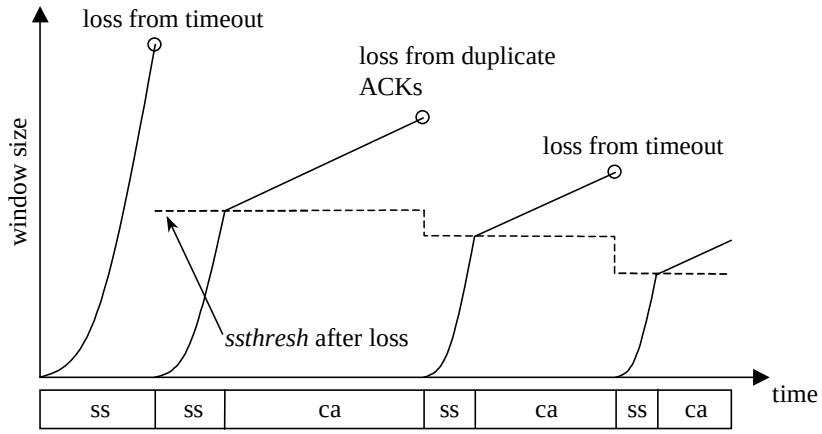

Figure 3.1: Window size behavior as a function of time in TCP Tahoe.

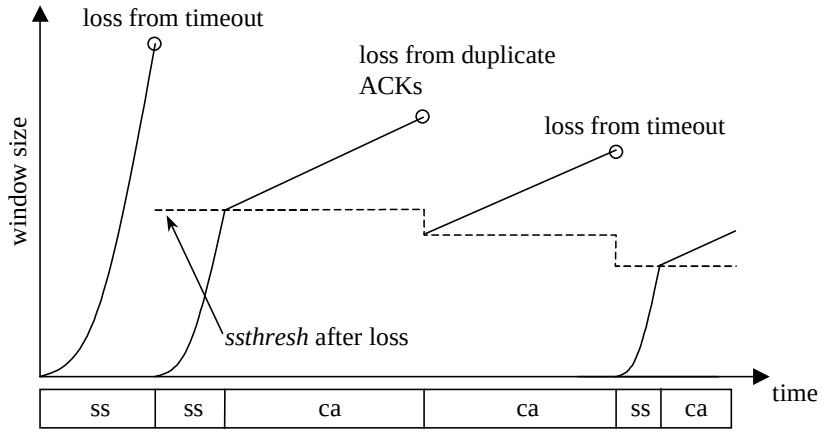

Figure 3.2: Window size behavior as a function of time in TCP Reno.

What this implies is that when a TCP connection detects a packet loss via duplicate ACKs it simply halves its congestion window and proceeds from there on as in congestion avoidance, i.e., increasing its congestion window linearly. This mechanism of avoiding unnecessary slow starts has been shown to increase the throughput of TCP connections considerably in the case of rare losses and large windows. The details of how this is achieved are omitted here and can be found, e.g., in [72]. If the packet loss is detected by a time out, the TCP connection is considered to be dead (i.e., there are no packets in transit to maintain the self clocking) and the connection performs a slow start.

An example is shown in Figure 3.1, how the window size behaves as a function of time in TCP Tahoe. The main observation is that the TCP connection always falls back to slow start, when there is a packet loss. The amount of time the connection is in the slow start state (ss in the picture) and the congestion avoidance state (ca in the picture) are also shown below the horizontal time axis of the picture. The same scenario is depicted for TCP Reno in Figure 3.2. From the picture it can be noticed how TCP Reno avoids one slow start, in this example.

3.3 The RED Algorithm

The idea of the RED algorithm is to provide feedback signals back to the TCP sources in the form of discarded (or marked) packets. This is done such that when the load of the queue (measured with an exponentially averaged queue length) starts to increase, the probabilistic packet dropping of RED will cause some of the TCP sources (but not all) to back off, resulting in a decrease in the packet arrival rate to the queue. The randomization of the packet dropping is the main reason why RED eliminates the problem of the so called global synchronization. The synchronization can occur when a buffer overflow takes place in a congested buffer, in which case all TCP sources observing a packet loss drop their sending rates at the same time and start increasing their sending rates at the same time until overflow occurs again, resulting in oscillations in the load of the queue. Additionally, RED helps in preventing bursty connections from suffering more packet loss than smoother behaving connections [50].

The RED algorithm works as follows. Consider a queue with a finite buffer which can hold up to K packets at a time. Let q be the queue length and avg be the exponentially averaged queue length (to be defined below) at a packet arrival instant. Also, c is the counter representing the time (in packet arrivals) since the previous packet discard. The RED algorithm randomly drops packets with probability p in the following manner.

For each arriving packet, the current averaged queue length avg is computed with

$$\begin{cases} avg \leftarrow (1-\beta)avg + \beta q, & \text{if } q > 0, \\ avg \leftarrow (1-\beta)^m avg, & \text{if } q = 0, \end{cases}$$

where $0 < \beta < 1$ is an appropriate constant, and m is the ratio of the time since the previous packet departure (i.e., the time instant when the system last became idle) and a typical transmission time for a small packet. The latter update equation is used to correct for the fact that by only using the first update equation the averaged queue length would not come down during idle periods, which can even be relatively long. Thus, with the latter update equation the averaged queue length is decreased by a factor representing how many small sized packets could have been sent during the idle period. If the buffer is full, the packet is lost (i.e., $p = 1$) and $c \leftarrow 0$. If $avg < T_{\min}$, the arriving packet is accepted (i.e., $p = 0$) and $c \leftarrow -1$. If $avg > T_{\max}$ the packet is discarded (i.e., $p = 1$) and $c \leftarrow 0$. However, if $T_{\min} \leq s_n \leq T_{\max}$, c is incremented by $c \leftarrow c + 1$, and the discarding probability is determined from

$$\begin{aligned} p_{ini} &= p_{\max}(avg - T_{\min})/(T_{\max} - T_{\min}), \\ p &= \min[1, p_{ini}/(1 - cp_{ini})]. \end{aligned}$$

Then, with probability p, the packet is discarded and $c \leftarrow 0$, otherwise it is accepted into the queue. The actions of the RED algorithm are summarized in Table 3.1. Typically, β is chosen to be relatively small, e.g., Floyd and Jacobson [50] propose using $\beta \geq 0.001$ and use $\beta = 0.002$ in their simulations. Recommendations for choosing the other parameters can be found in Floyd [51], where it is proposed that $p_{\max} \leq 0.1, T_{\min} \geq 5$ and $T_{\max} \geq 3 \times T_{\min}$.

	$avg < T_{\min}$	$T_{\min} \leq avg \leq T_{\max}$	$avg > T_{\max}$
$q < K$	accept	discard with prob. p	discard
$q = K$	discard	discard	discard

Table 3.1: Actions of the RED algorithm.

The role of the counter c in the RED algorithm is to distribute the packet drops of RED more evenly. In fact, as shown in [50], the use of the counter results in that the TCP sources observe a smoother flow of roughly uniformly spaced congestion signals, as opposed to what would happen if the discarding probability was given by p_{ini}.

The RED drop function, as defined by p_{ini}, of the original RED algorithm is depicted in Figure 3.3 (left picture). It consists of a linear increase in the dropping probability from 0 to $p_{\max}$ in the range $[T_{\min}, T_{\max}]$. At $T_{\max}$ the dropping probability has a discontinuity where p_{ini} jumps from $p_{\max}$ to 1. To avoid strong oscillations from occurring when load increases so high that the threshold $T_{\max}$ is reached, Floyd [53] has presented a more "gentle" variant of the RED algorithm, where the drop function is changed such that instead of the discontinuity at $T_{\max}$ the drop probability is linearly increased to 1 in the range $[T_{\max}, 2T_{\max}]$, as shown in Figure 3.3 (right picture).

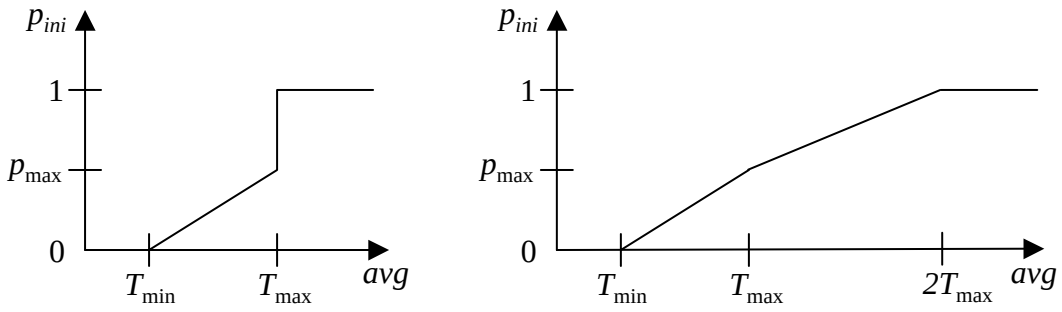

Figure 3.3: The drop function of the original RED algorithm (left picture) and the "gentle" variant of RED (right picture).

3.4 System Model for TCP-RED Interaction

RED has been designed to work in cooperation with TCP sources transmitting packets through the RED controlled buffer. Thus, a full description of the TCP-RED congestion control system consists of a closed system with two interacting parts (see Figure 3.4):

a) A queue receiving an aggregate flow of packets and reacting dynamically to the changes in the total flow rate under the control of the RED algorithm.

b) A population of TCP sources, each of which reacts to the packet losses at the queue as determined by the TCP flow control mechanism.

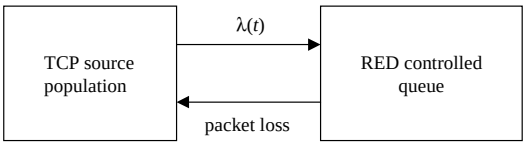

Figure 3.4: Interaction between the TCP population and the RED controlled queue.

3.4.1 Idealizations Used in Modeling

For the TCP part the key assumption is that the TCP sources are always in the congestion avoidance phase and time outs are not taken into account. Thus, for successfully transmitted packets the congestion window $w_i(t)$ of source i at time t is increased linearly and for each dropped packet $w_i(t)$ is halved. This is a standard assumption in TCP literature, see, e.g., [63], [52], [59], [65], [38] and [39]. In practice this implies that the model should capture the essential behavior of TCP Reno, in the case when the packet loss events are relatively rare and the delay-bandwidth product is high enough such that time outs do not occur [65]. An additional assumption is that the round trip time, denoted by R, is assumed to be a constant (i.e., the random queuing delays are not modeled) and the same for all sources.

For modeling the RED controlled buffer, a simplified version of the original RED algorithm is used, which is typical in RED analyzes and also adopted, e.g., in [66]. To be specific, the following simplifications are made. The case where the exponentially weighted average (EWA) queue length is computed differently when the arrival occurs into an empty queue is not considered. However, as the primary interest lies in heavily loaded queues, the effect of this special handling is of less importance. Additionally, the counter, which measures the time (in terms of packet arrivals) since the previous packet drop, is not modeled. (Note that the system with the counter can be approximated by a counterless system with the value for the RED parameter $p_{\max}$ doubled, see, e.g., [50].)

The simplified RED algorithm works as follows: Consider a queue with a finite buffer which can hold up to K packets at a time. Let q_n be the queue length at the n^{th} arrival. For each arriving packet, the EWA queue length at the n^{th} arrival, s_n, is computed with

$$s_n = (1 - \beta)s_{n-1} + \beta q_n, \tag{3.1}$$

where $0 < \beta < 1$ is the weighting parameter (as earlier). If the buffer is full, the packet is lost. If $s_n < T_{\min}$, the arriving packet is accepted; if $s_n > T_{\max}$ the packet is discarded. In the intermediate range $T_{\min} \leq s_n \leq T_{\max}$, the packet is discarded with probability

$$p(s_n) = \frac{p_{\max}(s_n - T_{\min})}{T_{\max} - T_{\min}},$$

and accepted otherwise.

3.4.2 The TCP Model

The TCP model used in the present work in Publication 7 is essentially the same as used by Kelly in [59], however, its derivation has been altered to get a clearer idea of the approximations involved. Consider m identical TCP sources with constant round trip time R. To simplify the notation, the delays in the time arguments are temporarily neglected and the equations are derived as if the changes to the window sizes occurred instantaneously at packet transmission epochs. The TCP aggregate has a window $W(t) = \sum_i w_i(t)$. The arrival rate (throughput), $\lambda_i(t)$, of source i is

$$\lambda_i(t) = \frac{w_i(t)}{R}.$$

Consider a small time interval $(t, t + \Delta t)$, or Δt for brevity. Let $A_i(\Delta t)$ denote the event that there is exactly 1 arrival from source i during Δt. Additionally, given that there is a packet arrival in Δt from source i, let $L_i(t)$ denote the event that this packet is lost and $L_i^c(t)$ its complement. The basic assumption on the traffic generated by a TCP source is that it is periodic with time dependent periods, but for each TCP source the packet transmission phase is random. This implies that given the current window size $w_i(t)$ of source i, the probability of an arrival in Δt is

$$\mathrm{P}\{A_i(\Delta t) \,|\, w_i(t)\} = \lambda_i(t)\Delta t = w_i(t)\Delta t/R,$$

and the probability of having more than 1 arrival in Δt equals zero. In TCP congestion avoidance, for each successfully transmitted packet source i increases its transmission window by $1/w_i(t)$, and for each lost packet the window is halved. The expectation of the change in the aggregate arrival rate $\lambda(t) = \sum_i \lambda_i(t)$, $\mathrm{E}[\Delta\lambda(t)]$, can be computed by conditioning on the set of window sizes $\{w_i(t)\}$ and $A_i(\Delta t)$ as follows

$$\begin{aligned}
\mathrm{E}[\Delta\lambda(t)] &= \frac{1}{R}\mathrm{E}\left[\mathrm{E}\left[\Delta W(t) \,|\, \{w_i(t)\}\right]\right] = \frac{1}{R}\mathrm{E}\left[\mathrm{E}\left[\sum_i \Delta w_i(t) \Big| \{w_i(t)\}\right]\right] \\
&= \frac{1}{R}\mathrm{E}\left[\sum_i \mathrm{E}\left[\Delta w_i(t) | \{w_i(t)\}, A_i(\Delta t)\right] \cdot \mathrm{P}\{A_i(\Delta t) \,|\, \{w_i(t)\}\}\right] \\
&= \frac{1}{R}\mathrm{E}\left[\sum_i \left(\mathrm{P}\{L_i^c(t) \,|\, \{w_i(t)\}\}\frac{1}{w_i(t)} - \mathrm{P}\{L_i(t) \,|\, \{w_i(t)\}\}\frac{w_i(t)}{2}\right)\frac{w_i(t)}{R}\Delta t\right] \\
&= \frac{\Delta t}{R^2}\mathrm{E}\left[\sum_i \left(\mathrm{P}\{L_i^c(t) \,|\, \{w_i(t)\}\} - \mathrm{P}\{L_i(t) \,|\, \{w_i(t)\}\}\frac{w_i^2(t)}{2}\right)\right].
\end{aligned}$$

If it is further assumed that the $w_i(t)$ processes are independent of each other and that the number of TCP sources is large, each individual source experiences a packet loss as determined by the queue receiving the aggregate traffic. The conditional probability $\mathrm{P}\{L_i(t) \,|\, \{w_i(t)\}\}$ corresponds then to the loss probability in the queue at time t, which is

denoted here by $P_L(t)$. Proceeding with the derivation, one obtains

$$\begin{aligned} \mathrm{E}[\Delta\lambda(t)] &\approx \frac{\Delta t}{R^2}\mathrm{E}\left[\sum_i \left((1-P_L(t)) - P_L(t)\frac{w_i^2(t)}{2}\right)\right] \\ &= \frac{m\Delta t}{R^2}\left((1-P_L(t)) - P_L(t)\frac{\mathrm{E}[w_i^2(t)]}{2}\right). \end{aligned}$$

To evaluate the term $\mathrm{E}[w_i^2(t)]$ the following approximation is used,

$$\mathrm{E}[w_i^2(t)] \approx \mathrm{E}[w_i(t)]^2 \approx \mathrm{E}[W(t)]^2/m^2.$$

Publication 7 discusses the minor influence of this approximation when compared with a situation where each $w_i(t)$ has a uniform distribution in the range $[2w/3, 4w/3]$. Thus, one obtains

$$\begin{aligned} \mathrm{E}[\Delta\lambda(t)] &\approx \frac{m\Delta t}{R^2}\left((1-P_L(t)) - P_L(t)\frac{\mathrm{E}[W(t)]^2}{2m^2}\right), \\ &= \frac{m\Delta t}{R^2}\left((1-P_L(t)) - P_L(t)\frac{\mathrm{E}[\lambda(t)]^2R^2}{2m^2}\right), \\ &= \left((1-P_L(t))\frac{m}{R^2} - P_L(t)\frac{\mathrm{E}[\lambda(t)]^2}{2m}\right)\Delta t. \end{aligned}$$

By the linearity of the Δ operator, $\mathrm{E}[\Delta\lambda(t)] = \Delta\mathrm{E}[\lambda(t)]$, denoting with $\mathrm{E}[\lambda(t)] = \bar{\lambda}(t)$, and letting $\Delta t \to 0$, one arrives at the ordinary differential equation (ODE)

$$\frac{d}{dt}\bar{\lambda}(t) = (1-P_L(t))\frac{m}{R^2} - P_L(t)\frac{\bar{\lambda}^2(t)}{2m}. \tag{3.2}$$

Note that solving (3.2) for the equilibrium, $d\bar{\lambda}(t)/dt = 0$, gives the familiar TCP steady state relation

$$\bar{\lambda} = \frac{m}{R}\sqrt{\frac{2(1-P_L)}{P_L}} \approx m\sqrt{2}\cdot\frac{1}{R\sqrt{P_L}},$$

where the last approximation holds for small P_L.

3.4.3 The Queue Model

By using the earlier assumptions on a large TCP population and random phases, the aggregate stream can be approximated by a time varying inhomogeneous Poisson process with rate $\lambda(t)$. Strictly speaking, $\lambda(t)$ is itself a stochastic process but it is assumed that the fluctuations are small and the approximation $\lambda(t) \approx \mathrm{E}[\lambda(t)]$ is used. Thus, the queue part of the model consists of a RED controlled queue receiving an inhomogeneous Poisson arrival stream with a deterministic time varying rate $\mathrm{E}[\lambda(t)] = \bar{\lambda}(t)$ governed by eq. (3.2). Additionally, the service times can have a general distribution with mean μ.

ODE for Averaged Queue Length

The aim is to develop a continuous time model for the transient behavior of the queue. The discrete time equation governing the time development of s_n is obtained from (3.1),

$$\Delta s_n = s_{n+1} - s_n = \beta(q_{n+1} - s_n). \tag{3.3}$$

Eq. (3.3) relates the values of s_{n+1} and s_n to each other at the arrival instants of the packets. However, the EWA queue length process can be studied in continuous time, i.e., the process $s(t)$. Consider a short interval of length Δt. Then by taking expectations of (3.3) and conditioning on the number of arrival events $A(\Delta t)$ occurring in the interval $(t, t + \Delta t)$ one obtains

$$\begin{aligned} \mathrm{E}[\Delta s(t)] &= \mathrm{E}\left[\Delta s(t) \,|\, A(\Delta t) = 1\right] \mathrm{P}\{A(\Delta t) = 1\} + \\ &\quad \mathrm{E}\left[\Delta s(t) \,|\, A(\Delta t) = 0\right] \mathrm{P}\{A(\Delta t) = 0\} + O((\Delta t)^2) \end{aligned}$$

from which

$$\begin{aligned} \Delta \mathrm{E}[s(t)] &= \beta \mathrm{E}[q(t) - s(t)] \bar{\lambda}(t) \Delta t + O((\Delta t)^2) \\ &= \bar{\lambda}(t) \beta \left(\mathrm{E}[q(t)] - \mathrm{E}[s(t)]\right) \Delta t + O((\Delta t)^2) \end{aligned}$$

Then, by denoting $\bar{s}(t) = \mathrm{E}[s(t)]$ and $\bar{q}(t) = \mathrm{E}[q(t)]$ and taking the limit $\Delta t \to 0$ one obtains the ODE

$$\frac{d}{dt}\bar{s}(t) = \bar{\lambda}(t)\beta(\bar{q}(t) - \bar{s}(t)) \tag{3.4}$$

Quasi-stationary Approximation for Averaged Queue Length

The exact dynamics of the term $\bar{q}(t)$ in (3.4) are unknown. To overcome this, in [71] a quasi-stationarity approximation, which is exact in the limit $\beta \to 0$, was made. The approximation is based on the fact that in practice we are interested in evaluating (3.4) for systems where β is small, say 10^{-3}. Then $s(t)$ changes very slowly compared to $q(t)$ and, hence, it can be assumed that the term $\bar{q}(t)$ is sufficiently well approximated by the mean stationary queue length of a hypothetical queue length process where the current value of the EWA queue length $s(t)$ controlling the access to the queue is fixed. To be precise, consider the stationary queue length $q^{(s)}$ as a function of the access control parameter s which is constant. The approximation amounts to $\bar{q}(t) = \mathrm{E}[q^{(s)}]|_{s=\bar{s}(t)}$, i.e., the time dependence is introduced parametrically to the stationary distribution and $\bar{q}(t)$ is thus given by the mean queue length of a M/G/1/K queue with arrival rate $\bar{\lambda}(t)(1 - p(\bar{s}(t)))$.

ODE for Instantaneous Queue Length

However, the simulation results in [71] showed that even when $\beta = 10^{-3}$, the finite time before the queue length process reaches stationarity has a noticeable effect on the queue dynamics. Thus, the expectation of the instantaneous queue length $\bar{q}(t)$ is analyzed here in more detail, and another ODE for $\bar{q}(t)$ is introduced. Therefore, we consider again a short interval of time Δt, condition on having 0, 1 or more arrivals and take the limit $\Delta t \to 0$. By using some additional assumptions the following result is obtained,

$$\frac{d}{dt}\bar{q}(t) = \bar{\lambda}(t)\Big(1 - \pi_K(\bar{q}(t))\Big)\Big(1 - p(\bar{s}(t))\Big) - \mu\Big(1 - \pi_0(\bar{q}(t))\Big), \tag{3.5}$$

where the first term is the expected rate at which packets are admitted to the queue and the second term is the expected rate at which packets leave the system.

In order to define the quantities $\pi_0(\bar{q}(t))$ and $\pi_K(\bar{q}(t))$ in (3.5) and to see the nature of the approximations involved, let us examine more closely the first term in (3.5). To correct the slight inaccuracy and abuse of notation of Publication 6, the justifications given here differ slightly from the one given in the publication. The queue length process $q(t)$ varies rapidly on a time scale that is short in comparison with other relevant time scales. Thus, it can be assumed that $q(t)$ roughly behaves as the stationary queue length of the M/G/1/K queue with mean $m(t)$, which itself is a random process on a slower time scale. Now, aside for the factor $\bar{\lambda}(t)$, the other factors in the first term of (3.5) represent the following expected acceptance probability $P_{\text{acc}}(t)$ of a packet at time t:

$$P_{\text{acc}}(t) = \mathrm{E}[1_{q(t)<K} \cdot R(t)],$$

where $R(t)$ denotes the Bernoulli variable describing whether RED will accept the packet or not. The expectation can be evaluated by conditioning on the values of $m(t)$ and $s(t)$,

$$\begin{aligned} \mathrm{E}[1_{q(t)<K} \cdot R(t)] &= \mathrm{E}\left[\mathrm{E}\left[1_{q(t)<K} \cdot R(t) \,|\, m(t), s(t)\right]\right], \\ &= \mathrm{E}\left[\Big(1 - \pi_K(m(t))\Big)\Big(1 - p(s(t))\Big)\right], \end{aligned}$$

where use has been made of the facts that the RED drop probability is only a function of $s(t)$ and that given $m(t)$ the queue length process $q(t)$ is independent of $s(t)$. It is not unreasonable to assume that the distributions of $m(t)$ and $s(t)$ are narrow and thus these random variables can approximately be replaced with their means, $\bar{q}(t)$ and $\bar{s}(t)$, in the above formula

$$\mathrm{E}\left[\Big(1 - \pi_K(m(t))\Big)\Big(1 - p(s(t)\Big)\right] \approx \Big(1 - \pi_K(\bar{q}(t))\Big)\Big(1 - p(\bar{s}(t))\Big).$$

In the above, the function $\pi_K(\bar{q}(t))$ denotes the stationary packet loss probability in an M/G/1/K queue with mean $\bar{q}(t)$ and a method is needed to efficiently construct this function (and for $\pi_0(\bar{q}(t))$, as well). To this end, let $\pi_k^\infty(\rho)$ denote the steady state probability of state k in the infinite capacity M/G/1 system with load ρ. The queue length distribution of the M/G/1 system is given by the well known Pollaczek-Khintchine formula [54, p. 230] and the distribution can be computed algorithmically as presented in, e.g., [74, p. 266]. Then the steady state probability of state k, $\pi_k(\rho)$, in the finite system can be obtained from the following relations (see [54, p. 230])

$$\begin{aligned} \pi_j(\rho) &= \frac{\pi_j^\infty(\rho)}{\pi_0^\infty(\rho) + \rho G(K)}, \quad j = 0, \ldots, K-1, \\ \pi_K(\rho) &= 1 - \frac{G(K)}{\pi_0^\infty(\rho) + \rho G(K)}, \\ G(K) &= \sum_{j=0}^{K-1} \pi_j^\infty(\rho). \end{aligned}$$

Additionally, it holds trivially that

$$\bar{q}(\rho) = \sum_{j=0}^{K} j\pi_j(\rho).$$

The steady state functions $\pi_0(\bar{q})$ and $\pi_K(\bar{q})$ are computed by eliminating ρ from the above relations. This technique is also similar to the approximations for nonstationary queues given in [75]. In practice, e.g., the function $\pi_0(\bar{q})$ is best derived by computing pairs of values $(\bar{q}(\rho), \pi_0(\rho))$ for a suitably dense discretization of ρ providing a discretized version of $\pi_0(\bar{q})$. Then one can use interpolation between the discretization points to compute $\pi_0(\bar{q})$ for any value of $\bar{q}$.

In Publication 7 the M/D/1/K system is used as an example. In Publication 6, the M/M/1/K queue is used, in which case the equations for constructing $\pi_0(\bar{q})$ and $\pi_K(\bar{q})$ are simply

$$\begin{cases} \pi_0 &= \dfrac{1}{1+\cdots+\rho^K}, \\ \bar{q} &= \displaystyle\sum_{n=0}^{K} \frac{n\rho^n}{1+\cdots+\rho^K}, \end{cases} \qquad \begin{cases} \pi_K &= \dfrac{\rho^K}{1+\cdots+\rho^K}, \\ \bar{q} &= \displaystyle\sum_{n=0}^{K} \frac{n\rho^n}{1+\cdots+\rho^K}. \end{cases}$$

Note that if the buffer is assumed to be infinite, (3.5) becomes

$$\frac{d}{dt}\bar{q}(t) = \bar{\lambda}(t)\Big(1 - p(\bar{s}(t))\Big) - \mu\Big(1 - \pi_0(\bar{q}(t))\Big).$$

The above approximation then gives in the M/M/1 case

$$\begin{cases} \pi_0 &= 1-\rho, \\ \bar{q} &= \dfrac{\rho}{1-\rho}, \end{cases}$$

yielding the result $\pi_0(\bar{q}) = 1/(1+\bar{q})$ and

$$\frac{d}{dt}\bar{q}(t) = \bar{\lambda}(t)\Big(1 - p(\bar{s}(t))\Big) - \mu\frac{\bar{q}(t)}{1+\bar{q}(t)}.$$

In the case of the M/G/1 system, one needs to solve

$$\begin{cases} \pi_0 &= 1-\rho, \\ \bar{q} &= \rho + \dfrac{\rho^2(1+C^2)}{2(1-\rho)}, \end{cases}$$

where C^2 is the squared coefficient of variation of the packet length, giving

$$1 - \pi_0 = \frac{\bar{q} + 1 - \sqrt{\bar{q}^2 + 2C^2\bar{q} + 1}}{1 - C^2},$$

and

$$\frac{d}{dt}\bar{q}(t) = \bar{\lambda}(t)\Big(1 - p(\bar{s}(t))\Big) - \mu\frac{\bar{q}(t) + 1 - \sqrt{\bar{q}(t)^2 + 2C^2\bar{q}(t) + 1}}{1 - C^2}.$$

3.4.4 The TCP-RED System Model

By combining eqs. (3.2), (3.4) and (3.5) one gets a model for the TCP-RED interaction omitting the effect of link delays,

$$\begin{cases} \dfrac{d}{dt}\bar{s}(t) = \bar{\lambda}(t)\beta(\bar{q}(t) - \bar{s}(t)), \\ \dfrac{d}{dt}\bar{q}(t) = \bar{\lambda}(t)\Big(1 - \pi_K(\bar{q}(t))\Big)\Big(1 - p(\bar{s}(t))\Big) - \mu\Big(1 - \pi_0(\bar{q}(t))\Big), \\ \dfrac{d}{dt}\bar{\lambda}(t) = (1 - P_L(t))\dfrac{m}{R^2} - P_L(t)\dfrac{\bar{\lambda}^2(t)}{2m}, \end{cases} \tag{3.6}$$

$$P_L(t) = p(\bar{s}(t)) + \pi_K(\bar{q}(t)) - p(\bar{s}(t))\pi_K(\bar{q}(t)).$$

However, a more accurate model is obtained by including the link delays in the system (but queuing delays are not modeled and hence the delays are not time dependent). In the following model $\bar{\lambda}(t)$ denotes the expected aggregate TCP transmission rate at the source. The acknowledgments arrive at the source with the delay R causing the update action to the current sending rate, and the amount of update depends on the current rate. Also, the TCP population will adjust its current sending rate based on packet losses having taken place in time $R/2$ prior to the current time. Similarly, the packets sent by the TCP population arrive at the queue after the link delay of $R/2$ and hence the queue will observe the $R/2$ delayed sending rate. This gives rise to a retarded functional differential equation (RFDE, or a delay differential equation):

$$\begin{cases} \dfrac{d}{dt}\bar{s}(t) = \bar{\lambda}(t - R/2)\beta(\bar{q}(t) - \bar{s}(t)), \\ \dfrac{d}{dt}\bar{q}(t) = \bar{\lambda}(t - R/2)\Big(1 - \pi_K(\bar{q}(t)))(1 - p(\bar{s}(t))\Big) - \mu\Big(1 - \pi_0(\bar{q}(t))\Big), \\ \dfrac{d}{dt}\bar{\lambda}(t) = -\dfrac{P_L(t - R/2)}{2m}\bar{\lambda}(t)\bar{\lambda}(t - R) + (1 - P_L(t - R/2))\,\dfrac{m}{R^2}\dfrac{\bar{\lambda}(t - R)}{\bar{\lambda}(t)}, \end{cases} \tag{3.7}$$

$$P_L(t) = p(\bar{s}(t)) + \pi_K(\bar{q}(t)) - p(\bar{s}(t))\pi_K(\bar{q}(t)).$$

Here a symmetric delay structure has been used, but any other delay structure can be handled with obvious modifications. Also, note that in the TCP model, P_L denoted the probability of the packet loss observed by the TCP population. In the above system, packets may be dropped due to the RED admission control or the buffer overflow. Hence, the packet loss is approximated by $P_L = p(\bar{s}) + \pi_K(\bar{q}) - p(\bar{s})\pi_K(\bar{q})$. Additionally, observe that in the last differential equation in (3.7), $\bar{\lambda}(t)$ appears with two time arguments: $\bar{\lambda}(t-R)$ represents the update rate at time t, whereas $\bar{\lambda}(t)$ arises from the amount of change to the current rate.

The initial conditions of (3.7) at time t_0 are given by the functions $\bar{q}, \bar{s} : [t_0, R/2] \to \mathbb{R}$ and $\bar{\lambda} : [t_0, R] \to \mathbb{R}$. Also, note that when there are no packet losses, the slope of the linear increase in the throughput is slightly less than m/R^2 due to the term $\bar{\lambda}(t - R)/\bar{\lambda}(t)$, a fact also seen in the simulations.

3.4.5 Model Validation by Simulation

The implementation of the simulator for verifying the accuracy of the TCP-RED model (3.7) is discussed in Publication 7. Briefly, the simulation is carried out as follows. The sources model the behavior of a greedy FTP source population transmitting constant length packets. Hence, each TCP source i has its own congestion window w_i (as opposed to the model), and R is a common constant for all sources. The congestion window of each source is incremented by $1/w_i$ for each accepted packet and halved for each lost packet (based on delayed information of the acknowledgement and loss events). The source behaves as a fluid process with instantaneous rate w_i/R, and sends a packet at those instants of time where the cumulative fluid attains an integer value. Thus, the aggregate traffic into the queue consists of a superposition of periodic sources with a time varying sending rate.

The biggest difference in the behavior of the fluid sources and "real" TCP sources is that the fluid sources try to space the packets in a "smooth" manner, as opposed to what may happen with real TCP. If a real TCP source is able to send packets at a rate comparable to the link speed of the bottleneck link, the transmission pattern of the real TCP source is bursty in nature: As long as all transmitted packets go through, in congestion avoidance, the window is increased linearly and each time the value of the congestion window exceeds a new integer value, two packets are sent back to back (one for the ACK and one corresponding to the window size increase). Thus, after the, say, first ACK, a source would send two packets back to back. The (possible) queuing at the bottle neck will not introduce much extra delay between the two packets and the ACKs for the two packets will arrive closely spaced at the sender. It will again send two packets corresponding to the ACKs and one extra packet as a result of the window size increase, and so on. With the exponential window increase policy of slow start, the above behavior is even more pronounced. This bursty transmission continues until there is a packet loss. At that point a real TCP source halves its congestion window and waits until enough ACKs have been received such that the number of packets in transit becomes again less than the current congestion window size, and the source will continue sending new packets at the same rate as with what the ACKs are received. On the other hand, the fluid source will not stay "silent" for a while upon receiving a packet loss event. Instead, it continues the transmission but at half rate. The exact relationship between the behavior of a real TCP source and the fluid source is yet to be determined and remains as an issue for further study requiring NS2 (Network Simulator 2) based simulations, and maybe even new models.

3.4.6 Extensions

The above system model can be easily extended to cover the case of a RIO controlled buffer. Heterogeneous round trip times can be modeled by dividing the TCP population into subpopulations each having a constant R_i. These extensions have been discussed in Publication 7. Additionally, queue length dependent queuing delays can be taken into account in a similar fashion as in [66] by replacing R with a function, which depends on the

fixed propagation delay(s) R and the queuing delays,

$$R(\bar{q}) = R + \bar{q}(t)/C,$$

where C is the service rate.

3.5 Stability Analysis

In this section some results from the general theory of stability analysis of linear systems are summarized, first for systems that can be described by ODE systems and then for systems represented by RFDEs. RFDEs are systems, where the time derivative depends on time lagged values of the system state variables. The results have been used in Publications 6 and 8. In the case of both (3.6) and (3.7) the systems are inherently nonlinear. Thus, the principle for examining the local stability of the systems is based on making a linearization of the system around its equilibrium point.

3.5.1 Ordinary Differential Equations

General Theory

Consider a system of linear ODEs of the form, expressed in vector notation,

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t). \tag{3.8}$$

By using a trial $\boldsymbol{\xi} e^{zt}$ in (3.8) for the time dependence, where $\boldsymbol{\xi}$ denotes a constant vector, one is lead to the eigenvalue equation

$$\det(z\mathbf{I} - \mathbf{A}) = 0.$$

The stability of (3.8) is determined by the roots of the eigenvalue equation. For the stability of the system, two criteria can be defined:

1. Non-oscillatory requirement: The system (3.8) can be required to have a non-oscillatory solution, in which case the requirement translates to that all of the eigenvalues must be real valued and negative.

2. Asymptotic stability: For asymptotic stability imaginary roots are allowed, but it is required that all the roots have a strictly negative real part (i.e., purely imaginary solutions are not allowed). The asymptotic stability is then determined by the root having the greatest, i.e., least negative, real part, and the degree of damping in the oscillations for each oscillation cycle is described by the absolute value of the ratio of the real part and the imaginary part.

Application to TCP-RED

In Publication 6, a system was considered where the time dependent arrival rate $\bar{\lambda}(t)$ was given by an exogenous deterministic function, i.e., the system consisted only of eqs. (3.4) and (3.5),

$$\begin{cases} \dfrac{d}{dt}\bar{s}(t) = \bar{\lambda}(t)\beta(\bar{q}(t) - \bar{s}(t)), \\ \dfrac{d}{dt}\bar{q}(t) = \bar{\lambda}(t)\Big(1 - \pi_K(\bar{q}(t))\Big)\Big(1 - p(\bar{s}(t))\Big) - \mu\Big(1 - \pi_0(\bar{q}(t))\Big), \end{cases} \tag{3.9}$$

The idea is to apply similar concepts as are used in the traditional control theory of deterministic systems. Therein one important consideration is the so called step response, i.e., the response of the system when the input changes as a step function. It is often required that the step response reaches equilibrium nicely, i.e., that the trajectories reach equilibrium without any oscillations or that the oscillations are dampened sufficiently quickly, as discussed above.

For the stability analysis, (3.9) is linearized around the equilibrium, when $\bar{\lambda}(t)$ is a step function

$$\bar{\lambda}(t) = \begin{cases} 0, & t \leq 0, \\ \bar{\lambda}, & t > 0. \end{cases}$$

Also it is assumed that the system state is $\bar{q} = \bar{s} = 0$ at time 0. The equilibrium is obtained by equating the right hand side of the ODE system (3.9) to 0,

$$\begin{cases} f_1(s^*, q^*) &= 0, \\ f_2(s^*, q^*) &= 0, \end{cases}$$

where $f_1(\cdot,\cdot)$ and $f_2(\cdot,\cdot)$ denote the right hand sides of the ODEs for $\bar{s}(t)$ and $\bar{q}(t)$, respectively. From the first equation it immediately follows that $s^* = q^*$. Finally, q^* is determined by the equation

$$f_2(q^*, q^*) = 0.$$

Close to the equilibrium (q^*, q^*), (3.9) can be approximated by its linearized version (expressed in vector notation)

$$\frac{d}{dt}\mathbf{x} = \mathbf{J}(q^*)\mathbf{x}, \tag{3.10}$$

where $\mathbf{x} = [\bar{s}, \bar{q}]^{\mathrm{T}}$ and $\mathbf{J}(q^*)$ is the Jacobian evaluated at the equilibrium,

$$\mathbf{J}(q^*) = \left(\begin{array}{cc} \dfrac{\partial f_1(\bar{s},\bar{q})}{\partial \bar{s}} & \dfrac{\partial f_1(\bar{s},\bar{q})}{\partial \bar{q}} \\ \dfrac{\partial f_2(\bar{s},\bar{q})}{\partial \bar{s}} & \dfrac{\partial f_2(\bar{s},\bar{q})}{\partial \bar{q}} \end{array} \right) \Bigg|_{\bar{s}=q^*, \bar{q}=q^*}.$$

The stability of (3.10) is determined by the eigenvalues of $\mathbf{J}(q^*)$. By using the following short hand notation for the elements of $\mathbf{J}(q^*)$,

$$\mathbf{J}(q^*) = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}.$$

the eigenvalue equation reads

$$z^2 - (a_1 + a_4)z + a_1 a_4 - a_2 a_3 = 0. \tag{3.11}$$

The eigenvalues are, in this case, obtained as a solution to a quadratic equation and, hence, both eigenvalues are at the same time either real or complex. The possible stability criteria are:

1. Non-oscillatory requirement: Since (3.11) is a quadratic equation, its roots are real valued if the discriminant of (3.11) is positive.
2. Asymptotic stability: If the roots of (3.11) are imaginary, they are always of the form $x \pm iy$, where i denotes the imaginary part and the degree of damping is governed by $\text{abs}(x/y)$, i.e., the greater the value of $\text{abs}(x/y)$ the more the oscillations are damped for each oscillation cycle.

Note that the above method can also be applied to (3.6). In that case $\mathbf{J}(\lambda^*, q^*)$ is a 3×3 matrix and the characteristic equation is a third order polynomial.

3.5.2 Delay Differential Equations

General Theory

Here the stability analysis related to RFDEs is discussed briefly (for details see [46, 55]). The theory for RFDEs parallels the one for ODEs. Consider a vector equation

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \sum_i \mathbf{B}_i \mathbf{x}(t - \tau_i),\ \tau_i > 0\ \ \forall i. \tag{3.12}$$

The role of eigenvalues for ODEs is replaced by the roots of the characteristic equation

$$\det \Delta(z) = \det \left(z\mathbf{I} - \mathbf{A} - \sum_i e^{-z\tau_i} \mathbf{B}_i \right) = 0$$

and the solution of (3.12) is asymptotically characterized by $e^{z_j t}$-terms, where $\det \Delta(z_j) = 0$. Contrary to polynomial eigenvalue equations for ODEs, the characteristic equation can have infinitely many roots (but only finitely many in the right half-plane (RHP)). The role of the initial condition with ODEs is replaced by the initial "history" and hence all stability results also change accordingly meaning that large oscillations in the initial history around an equilibrium may not be guaranteed to converge towards the equilibrium value (as would be with ODEs). Also recall that the methods based on linearizations give only local results.

The asymptotic stability of (3.12) is characterized by the absence of roots in the RHP or on the imaginary axis. However, in general, the numerical computation of the roots of the characteristic equation is not a suitable approach to study the stability. Sufficient conditions

for stability can be obtained using results on roots of certain characteristic equations, see [59, 73], also Lyapunov methods can be generalized to RFDEs to give sufficient conditions for stability or instability.

The approach used in this work, see [46] for a theoretical discussion, is to look for necessary and sufficient conditions for stability in a parameter space $(\alpha_1, \alpha_2) \in \mathbb{R} \times \mathbb{R}$, where α_1, α_2 are external parameters on which $\mathbf{A}$ and $\mathbf{B}$ (and possibly the τ_i) depend. The idea is then to study when the root of the characteristic equation passes the imaginary axis, causing non-decaying oscillatory solutions, and then unstable solutions when the root is in the RHP.

Given a characteristic equation $F(\alpha_1, \alpha_2, z) = 0$, which is analytic in z, let

$$\begin{aligned} G_1(\alpha_1, \alpha_2, x, y) &= \text{Re } F(\alpha_1, \alpha_2, x + iy), \\ G_2(\alpha_1, \alpha_2, x, y) &= \text{Im } F(\alpha_1, \alpha_2, x + iy). \end{aligned}$$

Suppose a point $(\alpha_1^0, \alpha_2^0, 0, y)$ has been found such that $G_1(\alpha_1^0, \alpha_2^0, 0, y) = G_2(\alpha_1^0, \alpha_2^0, 0, y) = 0$, i.e., there is a root on the imaginary axis. If the matrix

$$\mathbf{M} = \begin{pmatrix} \frac{\partial G_1}{\partial \alpha_1} & \frac{\partial G_1}{\partial \alpha_2} \\ \frac{\partial G_2}{\partial \alpha_1} & \frac{\partial G_2}{\partial \alpha_2} \end{pmatrix}_{(\alpha_1^0, \alpha_2^0, 0, y)}$$

is nonsingular, the equations

$$\begin{cases} G_1(\alpha_1, \alpha_2, 0, y) = 0, \\ G_2(\alpha_1, \alpha_2, 0, y) = 0, \end{cases}$$

have a locally unique solution curve $(\alpha_1(y), \alpha_2(y))$. Moreover, the roots with positive real parts are in the RHP of the parameter space to the left of curve $(\alpha_1(y), \alpha_2(y))$, when this curve is followed in the direction of increasing y, whenever $\det \mathbf{M} < 0$ and to the right when $\det \mathbf{M} > 0$. This allows us to detect the stable region in the parameter space in which there are no critical roots in the RHP.

Application to TCP-RED

To facilitate the stability analysis, (3.7) is simplified. First, the buffer is assumed to have an infinite size, in which case $P_L(t) = p(s)$, i.e., losses are only due to RED drops. This is reasonable as, in practice, the buffers of modern routers are very large and, moreover, the operating point of the system should anyway be such that it is the RED algorithm that performs the congestion control, not the packet drops caused by buffer overflow. Additionally, the quasi-stationarity approximation for $\bar{q}(t)$, introduced earlier, is used, where it is assumed that $\bar{q}(t)$ is approximated sufficiently well by the mean queue length of an M/G/1 queue receiving an arrival stream thinned by the factor $(1 - p(\bar{s}(t))$, i.e., the RED packet non-discard probability. Without loss of generality, assuming deterministic service times of duration 1, and using the above approximation for $\bar{q}(t)$ together with the assumption of an infinite queue length, $\bar{q}(t)$ can be related directly to the arrival rate $\lambda(t)$ via the well known Pollaczek-Khintchine formula

$$\bar{q}(\lambda) = \lambda + \frac{\lambda^2}{2(1 - \lambda)}.$$

Hence, the RFDE model becomes

$$\begin{cases} \dfrac{d}{dt}\bar{s}(t) = \bar{\lambda}(t-R/2)\beta\Big(\bar{q}((1-p(\bar{s}(t)))\bar{\lambda}(t-R/2)) - \bar{s}(t)\Big), \\ \dfrac{d}{dt}\bar{\lambda}(t) = -\dfrac{p(\bar{s}(t-R/2))}{2m}\bar{\lambda}(t)\bar{\lambda}(t-R) + (1-p(\bar{s}(t-R/2)))\,\dfrac{m}{R^2}\dfrac{\bar{\lambda}(t-R)}{\bar{\lambda}(t)}. \end{cases} \tag{3.13}$$

Let f and g denote the right-hand sides of (3.13) and let $(\bar{s}^*, \bar{\lambda}^*)$ denote the equilibrium, i.e., the solution of $f(\bar{s}^*, \bar{\lambda}^*) = g(\bar{s}^*, \bar{\lambda}^*) = 0$. To perform the linearization of (3.13) around $(\bar{s}^*, \bar{\lambda}^*)$, let $\bar{\lambda}_1 = \bar{\lambda}(t)$, $\bar{\lambda}_2 = \bar{\lambda}(t-R/2)$, $\bar{\lambda}_3 = \bar{\lambda}(t-R)$, $\bar{s}_1 = s(t)$ and $\bar{s}_2 = s(t-R/2)$, and (3.13) can be written as

$$\begin{cases} \dfrac{d}{dt}\bar{s}_1(t) = \bar{\lambda}_2\beta\Big(\bar{q}((1-p(\bar{s}_1))\lambda_2) - \bar{s}_1\Big) = f(\bar{s}_1, \bar{\lambda}_2), \\ \dfrac{d}{dt}\bar{\lambda}_1 = -\dfrac{p(\bar{s}_2)}{2m}\bar{\lambda}_1\bar{\lambda}_3 + (1-p(\bar{s}_2))\,\dfrac{m}{R^2}\dfrac{\bar{\lambda}_3}{\bar{\lambda}_1} = g(\bar{s}_2, \bar{\lambda}_1, \bar{\lambda}_3). \end{cases}$$

In vector notation $\mathbf{x}(t) = [\bar{s}(t), \bar{\lambda}(t)]^T$, the linearized system is of the form

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{x}(t-R/2) + \mathbf{C}\mathbf{x}(t-R),$$

where $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ are the Jacobian matrices

$$\mathbf{A} = \begin{pmatrix} \frac{\partial f}{\partial \bar{s}_1} & \frac{\partial f}{\partial \bar{\lambda}_1} \\ \frac{\partial g}{\partial \bar{s}_1} & \frac{\partial g}{\partial \bar{\lambda}_1} \end{pmatrix}_{(\bar{s}^*, \bar{\lambda}^*)} = \begin{pmatrix} a_{11} & 0 \\ 0 & a_{22} \end{pmatrix}, \qquad \mathbf{B} = \begin{pmatrix} \frac{\partial f}{\partial \bar{s}_2} & \frac{\partial f}{\partial \bar{\lambda}_2} \\ \frac{\partial g}{\partial \bar{s}_2} & \frac{\partial g}{\partial \bar{\lambda}_2} \end{pmatrix}_{(\bar{s}^*, \bar{\lambda}^*)} = \begin{pmatrix} 0 & b_{12} \\ b_{21} & 0 \end{pmatrix},$$

$$\mathbf{C} = \begin{pmatrix} 0 & 0 \\ 0 & \frac{\partial g}{\partial \bar{\lambda}_3} \end{pmatrix}_{(\bar{s}^*, \bar{\lambda}^*)} = \begin{pmatrix} 0 & 0 \\ 0 & c_{22} \end{pmatrix}.$$

In this case, at the equilibrium $(\bar{s}^*, \bar{\lambda}^*)$, the partial derivative $c_{22} = 0$ and, thus, the linearized equation reads

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{x}(t-R/2), \tag{3.14}$$

with the non-zero matrix elements being

$$\begin{aligned} a_{11}|_{(\bar{s}^*, \bar{\lambda}^*)} &= \bar{\lambda}^*\beta\Big(-\bar{\lambda}^* p'(\bar{s}^*)\bar{q}'(1-p(\bar{s}^*))\bar{\lambda}^*) - 1\Big) \le 0, \\ a_{22}|_{(\bar{s}^*, \bar{\lambda}^*)} &= -\frac{\bar{\lambda}^* p(\bar{s}^*)}{m} \le 0, \\ b_{12}|_{(\bar{s}^*, \bar{\lambda}^*)} &= \bar{\lambda}^*\beta(1-p(\bar{s}^*))\bar{q}'((1-p(\bar{s}^*))\bar{\lambda}^*) \ge 0, \\ b_{21}|_{(\bar{s}^*, \bar{\lambda}^*)} &= -p'(\bar{s}^*)\Big(\frac{(\bar{\lambda}^*)^2}{2m} + \frac{m}{R^2}\Big) \le 0. \end{aligned}$$

The characteristic equation of the linearized system (3.14) is

$$\det\left(\mathbf{A} + \mathbf{B}e^{-zR/2} - z\mathbf{I}\right) = \begin{vmatrix} a_{11} - z & b_{12}e^{-zR/2} \\ b_{21}e^{-zR/2} & a_{22} - z \end{vmatrix} = 0,$$

which results in

$$e^{zR}(a_{11} - z)(a_{22} - z) - b_{12}b_{21} = 0. \tag{3.15}$$

The requirement for the asymptotic stability is that in the parameter space of (3.15), i.e., the space consisting of the values of the tuple $(a_{11}, a_{22}, b_{12}b_{21})$, (3.15) has no roots with positive real parts. In Publication 8, it is shown how the stable parameter region is determined by taking a_{22} as a fixed parameter and then solving the stable region in the parameter space of $(b_{12}b_{21}, a_{11})$ and, also, how the results can be used to explore the stability boundaries of the system as functions of the physical system parameters.

3.6 Conclusions

The work in this chapter contains the main results from Publications 6-8. The publications constitute a progression where, by employing various stochastic assumptions and approximations, a dynamic model is derived to describe the interaction between an idealized homogeneous TCP population and a RED controlled buffer. The model is presented in Section 3.4 and it consists of a set of coupled differential equations governing the dynamic evolution of the expected values of the queue length processes (instantaneous and averaged) and the packet arrival rate. The main idea in the modeling is to derive a system that is able to capture the oscillations originally seen in the simulations in [71], and which is simple and accurate enough to be used for analyzing the stability of the system as a function of the system parameters. As discussed in Section 3.1.1, the modeling approach and the application of control system theory for examining the stability of the system is similar to the independent work in [66] and [56].

The derivation of the model is done in two separate parts: in the first part (Publication 6) the queue is considered alone receiving a packet stream characterized by an exogenous function giving the average packet arrival rate of packets into the queue, and in the second part (Publication 7), the interaction between the TCP sources and the queue is introduced along with the link delays to take into account the effect of the delay in the feedback loop. The numerical results of Publications 6 and 7 are intended to validate the model by comparing results obtained from simulations. In particular, the simulations of Publication 7 verify that the stochastic assumptions made in the model derivation are not so important. Additionally, Publication 6 provides initial results from applying control theoretic techniques in studying the stability of a RED controlled queue receiving a load determined by a step function. As a result, conditions are obtained such that the queue does not exhibit any oscillations.

The stability analysis of the delay differential equation system requires some further simplifications (infinite buffer size, omission of the equation for instantaneous queue length). With the aid of these, as shown in Publication 8 (and Section 3.5.2), it is possible to define a method whereby the stable parameter region of the system can be determined numerically and the stability of the system can be studied directly as a function of the system variables. This is illustrated by the numerical results in Publication 8.

In conclusion, it can be said that the model developed for the TCP-RED interaction is

versatile enough to give valuable insight into the many complex issues affecting its dynamics. Also, requiring stability of the expectations of the queue state variables gives one well defined and sensible way of solving the parameterization problem of the RED algorithm. However, a design procedure for ultimately solving the parameterization problem of real life Internet routers is still a part of future work. Additionally, as part of the future research, some issues related to the TCP model requiring further study include: The model of Publication 7 is verified by using sources having a TCP-like behavior. However, the relationship between the behavior of these and real TCP sources is not entirely clear (as alluded to already in Section 3.4.5). Also, the model assumes greedy sources that only operate in the congestion avoidance phase. The impact of stochastically arriving short-lived non-greedy connections (e.g., connections with finite file sizes or intermittent transmissions) should be investigated. This requires modeling the slow-start phase, as well. Finally, the model assumes a constant round trip time for all sources. In practice, the round trip times vary from one connection to another and the impact of this should be investigated thoroughly.

Chapter 4

Summaries of Publications and Author's Contributions

4.1 Publication 1

In Publication 1, the use of the Gibbs sampler is investigated. The first contribution of the publication is the identification of a suitable partitioning of the state space of the system that makes the application of the Gibbs sampler easy in the case of the multiservice loss system. The Gibbs sampler then provides an alternative to the traditional MC rejection method for generating the samples. Two approaches are considered: one based on generating a Markov chain with the correct steady state distribution of the system and another by which it is possible to generate a Markov chain with a uniform steady state distribution. As a second contribution, the first results in variance reduction methods are given, where the method of conditional expectations is applied. The method is formulated by using as the conditional expectation function the contribution that is obtained when making "virtual" transitions from a given state of the Markov chain generated by the Gibbs sampler, and computing the conditional expectation of these "virtual" transitions hitting the set of blocking states. The application of this is shown both in the case of using the correct steady state distribution and the uniform distribution. The numerical results show that the results obtained by using the Gibbs sampler combined with the conditional expectations compare favorably with other known methods (viz., the heuristics of Ross) for estimating the blocking probabilities.

4.2 Publication 2

Publication 2 presents the generalization of the method of conditional expectations to the case where the samples can be generated with any method, i.e., where the samples are not necessarily generated with the Gibbs sampler, as is assumed in Publication 1. A general problem of estimating the expectation of a multidimensional random variable is considered

first. In the case of i.i.d. samples, it is shown analytically that by using the method of conditional expectations and an appropriate conditioning function, the variance of the resulting estimator is indeed smaller than that of the basic MC estimator. Then the application of the results in the context of the multiservice loss system is given, where a suitable partitioning is identified. The amount of variance reduction obtained with this method is demonstrated with some numerical examples.

4.3 Publication 3

In Publication 3 the first results on applying IS are reported. The main contribution of the paper is a novel IS distribution to be used for estimating the blocking probabilities in multiservice loss systems. The IS distribution has a composite form consisting of a weighted combination of exponentially twisted distributions. In the literature it has been shown that an asymptotically efficient IS distribution is of the composite form for certain types of problems suggesting that a composite distribution leads to an efficient sampling distribution even in the non-asymptotic regime. Further heuristic justification for this choice is also provided by the analysis of the observed variable when using IS. The result from the analysis shows that to have a control of the variance of the observed variable, it is important to keep the value of the likelihood ratio as constant as possible. This heuristics also provides a rule for fixing the weights of the composite distribution, which are left open in the asymptotic theory. By using the composite IS distribution, it is possible to efficiently sample all the blocking states related to the links that the considered traffic class uses. The numerical experiments confirm the efficiency of the proposed method over other methods available in the literature.

4.4 Publication 4

The work on efficient IS methods for estimating the blocking probabilities in multiservice loss systems is continued in Publication 4, where, instead of using exponentially twisted distributions, a different approach is introduced. The first contribution of the paper is the derivation of the decomposition result for the multiservice loss system, allowing the original estimation problem to be decomposed into independent simpler subproblems, each roughly corresponding to estimating the blocking probability contribution from a single link. The main contribution of the publication is the derivation of two novel importance sampling distributions, which very closely approximate the ideal importance sampling distribution for each subproblem. In both methods, the idea is to try to generate samples directly into the blocking state region. The difference between the methods is that the first method, the inverse convolution method, achieves this exactly, while the second one, using a fitted Gaussian distribution, only approximately. Algorithmic presentations of both methods are also given, and, especially, the algorithm for the inverse convolution method can be considered to be one of the main contributions of this thesis. Finally, the paper gives a

dynamic control algorithm for optimally allocating the samples between different subproblems. The numerical results demonstrate that the variance reduction obtained with the methods, especially with the inverse convolution method, is indeed remarkable surpassing the efficiency of all previously reported methods, including the results for the composite distribution approach of Publication 3.

4.5 Publication 5

In Publication 5 the application of the decomposition principle and the inverse convolution method is shown for estimating the blocking probabilities in a multicast loss system. The multicast loss system is similar to the multiservice loss system studied in Publications 1–4, except that the state of the system is defined in a different manner. In the multicast loss system, the state of a given link does not correspond to a unique global state of the network. Instead, the mapping between a given link state and the network state is surjective, implying that there are several possible network states resulting in the same link state. The main contribution of the paper is the derivation of the algorithm for the inverse convolution method for the multicast loss system. The state representation results in an inverse convolution algorithm with two steps: in the first step the state of a given link is generated under a suitable conditioning, and in the second step, the global state is generated using a similar inverse convolution algorithm subject to the condition of the given link state. The numerical examples demonstrate the outstanding performance of the inverse convolution method also in this case. Additionally, in the numerical studies, the use of optimal allocation of the samples is compared with an implementation without optimal sample allocation, allowing the reuse of samples between the different subproblems.

4.6 Publication 6

Publication 6 considers the performance of a single queue where the simplified RED algorithm is used to control the access to the queue. The packet stream into the queue has been assumed to be Poisson with a time varying arrival rate, representing the aggregate traffic generated by a population of TCP sources. In the paper it is assumed that the arrival rate is determined by an exogenous function that does not depend on the queue variables, i.e., the feedback between the sending rate of the TCP population and the queue is not modeled. The main contribution of the paper is a dynamic model capturing the time dependent behavior of the expectations of the instantaneous queue length and the exponentially averaged queue length. The work builds on one of the ideas presented in Sharma et al. [71], where a differential equation is derived pertaining to the time evolution of the averaged queue length process. The simulations in [71], however, revealed oscillations in the instantaneous queue length which the model could not reproduce. Thus, in the present paper, another differential equation is introduced for characterizing the time evolution of the expectation of the instantaneous queue length. Also, new quasi-stationarity approximations are given for

approximating the unknown time dependent probabilities of the queue to be full and empty, respectively. Numerical results show that the model is able to capture well the behavior of the corresponding simulated system under different arrival rate functions. For instance, when using a step arrival rate function, the oscillations in the expectations are nicely reproduced. Note that the derivation of the differential equations is different from [71], since in the present paper the differential equations describe the expectations of the state variables, whereas in [71] the averaged queue length process is considered. Finally, initial results are provided from the application of control theoretic methods to obtain guidelines in choosing the parameters of the system. The response of the system is analyzed by linearizing the differential equations around the equilibrium point. By using a step function as an input, the stability of the system is characterized by the properties of the roots of the eigenvalue equation and it is possible to determine requirements, e.g., such that the system does not exhibit oscillations around the equilibrium point.

4.7 Publication 7

Publication 7 extends the work of Publication 6 by presenting a dynamic model for the TCP source population behavior and combining it with the queue model of Publication 6. The TCP model, however, represents an idealized version of TCP behavior, where the sources are assumed to always behave as in congestion avoidance, i.e., slow start (time outs) is not modeled. The TCP model is essentially the same as the one developed by Kelly (see [58]), but its derivation is altered to gain a better understanding of the approximations that are needed. The main contribution of the paper is the system of coupled differential equations for the population of TCP sources and the RED controlled queue, which also takes into account the delays in the feedback between the queue and the TCP source population. The system consists of three RDFEs describing the time evolution of the expectation of the arrival rate, the expectation of the instantaneous queue length and the expectation of the exponentially averaged queue length. In the queue model, the quasi-stationarity approximations of Publication 6 are generalized to the case of generally distributed packet lengths. The model can be used to explore, for example, parameter settings or stability surfaces (which are difficult to study by using simulations). Additionally the paper discusses various extensions of the model, e.g., the possibility to handle differentiated service classes with simple modifications (a model for a RIO controlled queue) and dealing with inhomogeneous round trip times. The emphasis of the paper is on a careful derivation of the model and the verification of its accuracy by simulations. In the simulation, none of the stochastic approximations of the model are used, and the very good agreement with the model shows that the stochastic assumptions are not important.

4.8 Publication 8

Publication 8 presents the stability analysis of the RFDE system, defined in Publication 7, describing the dynamics between a large TCP population and a RED controlled queue. To facilitate the analysis some simplifications are made to reduce the dimensionality of the system. In particular, the queue is assumed to have an infinite size and a quasi stationary approximation is used, obsoleting the equation for the instantaneous queue length. Thus, the RFDE system is expressed in terms of the expected values of the aggregate rate of the TCP population and the exponentially averaged queue length process. The main contribution of the paper is a technique whereby it is possible to derive, not only sufficient, but also necessary conditions for the stability. The technique is based on the analysis of the characteristic equation of the system, linearized around its equilibrium. Instead of searching for the roots of the characteristic equation, the parameter space of the characteristic equation is analyzed and the part of the parameter space is solved numerically, where the characteristic equation has no solutions with positive real parts, thus revealing the stable parameter region. This can easily be done numerically. Moreover, it is possible to compute numerically stability boundaries of the system as functions of the RED or TCP parameters. The resulting curves can be directly used for tuning of the RED queue such that the system is asymptotically stable.

4.9 Author's Contributions to Publications

In the publications relating to the fast simulation techniques for multiservice loss systems, i.e., Publications 1–4, the present author is also the first author. The main ideas and methods presented in the papers have emerged from an interactive process with Prof. Jorma Virtamo, where both parties have contributed to the results. The implementation of the different algorithms and the numerical evaluation of the methods has been done by the present author, except in Publication 4, where also Prof. Virtamo contributed. Publications 1–4 were written by the author under the guidance of Prof. Virtamo.

In Publication 5, the present author is the first author. The realization that it is possible to apply the decomposition result and the inverse convolution method to the multicast loss system originated from discussions between all three authors. The algorithm for testing the idea was implemented later by Mr. Jouni Karvo, Lic.Sc. (Tech.), in collaboration with the present author, who also performed the numerical experiments for verifying the efficiency of the method. The article itself was written as a joint effort involving all three authors.

In the publications on modeling and analysis of the TCP-RED interaction, i.e., Publications 6–8, the present author is the first author in Publication 6 and the second author in Publications 7 and 8. In the work for Publication 6, the oscillations were first observed in the simulations implemented and performed by the present author. The analytical model for describing the queue dynamics was developed in close cooperation between the present author and Prof. Virtamo. The numerical experiments and simulation verifications were

carried out by the present author. The paper was written by the present author under the guidance of Prof. Virtamo.

In Publication 7, the idea of applying the approach by Kelly et al. in the modeling of the TCP population emerged from discussions with Dr. Peter Key from Microsoft Research. The complete analytical model was developed as a joint group effort involving the present author, Dr. Pirkko Kuusela and Prof. Virtamo. The algorithms for evaluating the RFDE system were written by Dr. Kuusela with some support from the present author. The present author was also responsible for realizing the simulator which was used for validating the analytical model. The paper was mainly written by the three first authors, with Dr. Key providing some useful suggestions and comments during the preparation of the paper. In Publication 8, the used method and the theoretical background were provided by Prof. Virtamo and Dr. Kuusela. The algorithms were implemented for the most part by the present author with help from Dr. Pirkko Kuusela. The paper was mainly written by Dr. Kuusela and the present author.

Bibliography

Fast Simulation Techniques for Loss Systems

[1] A. A. Akyamaç, Z. Haraszti, J. K. Townsend, "Efficient Rare Event Simulation Using DPR for Multidimensional Parameter Spaces", in Proceedings of the 16^{th} International Teletraffic Congress, ITC 16, Edinburgh, United Kingdom, June 1999, pp. 767–776.

[2] G. L. Choudhury, K. K. Leung, W. Whitt, "An Algorithm to Compute Blocking Probabilities in Multi-Rate Multi-Class Multi-Resource Loss Models", Advances in Applied Probability, vol. 27, 1995, pp. 1104–1143.

[3] S. Chung, K. W. Ross, "Reduced Load Approximations for Multirate Loss Networks", IEEE Transactions on Communications, vol. 41, no. 8, 1993, pp. 1222–1231.

[4] M. Cottrell, J. Fort, G. Malgouyres, "Large Deviations and Rare Events in the Study of Stochastic Algorithms", IEEE Transactions on Automatic Control, vol. 28, no. 9, 1983, pp. 907–920.

[5] M. Devetsikiotis, J. K. Townsend, "Statistical Optimization of Dynamic Importance Sampling Parameters for Efficient Simulation of Communication Networks", IEEE Transactions on Networking, vol. 1, no. 3, 1993, pp. 293–305.

[6] D. G. Down, J. T. Virtamo, "Blocking Probabilities in Multirate Circuit Switched Networks: Capturing Dependent Behavior", in Proceedings of the 13^{th} Nordic Teletraffic Seminar, NTS 13, Trondheim, Norway, August 1996, pp. 18–28.

[7] P. J. Fleming, D. Schaeffer, B. Simon, "Efficient Monte Carlo Simulation of a Product Form Model for a Cellular System with Dynamic Resource Sharing", ACM Transactions on Modeling and Computer Simulation, vol. 5, no. 1, 1995, pp. 3-21.

[8] M. R. Frater, T. M. Lennon, B. D. O. Anderson, "Optimally Efficient Estimation of the Statistics of Rare Events in Queuing Networks", IEEE Transactions on Automatic Control, vol. 36, no. 12, 1991, pp. 1395–1405.

[9] P. Glasserman, P. Heidelberger, P. Shahabuddin, T. Zajic, "A Large Deviations Perspective on the Efficiency of Multilevel Splitting", IEEE Transactions on Automatic Control, vol. 43, no. 12, 1998, pp. 1666-1679.

[10] P. W. Glynn, D. L. Iglehart, "Importance Sampling for Stochastic Simulations", Management Science, vol. 35, no. 11, 1989, pp. 1367–1392.

[11] P. W. Glynn, "Likelihood Ratio Gradient Estimation for Stochastic Systems", Communications of the ACM, vol. 33, no. 10, 1990, pp. 75–84.

[12] A. Goyal, P. Shahabuddin, P. Heidelberger, V. F. Nicola, P. W. Glynn, "A Unified Framework for Simulating Markovian Models of Highly Dependable Systems", IEEE Transactions on Computers, vol. 41, no. 1, 1992, pp. 36–51.

[13] J. M. Hammersley, D. C. Handscomb, "Monte Carlo Methods", Methuen's Monographs on Applied Probability and Statistics, 1967, 168 p.

[14] P. E. Heegaard, "Efficient Simulation of Network Performance by Importance Sampling", in Proceedings of the 15^{th} International Teletraffic Congress, ITC 15, Washington DC, USA, June 1997, pp. 623–632.

[15] P. Heidelberger, "Fast Simulation of Rare Events in Queuing and Reliability Models", ACM Transactions on Modeling and Computer Simulation, vol. 5, no. 1, 1995, pp. 43–85.

[16] J. Y. Hui, "Resource Allocation for Broadband Networks", IEEE Journal on Selected Areas in Communications, vol. 6, no. 9, 1988, pp. 1598–1608.

[17] J. Karvo, J. Virtamo, S. Aalto and O. Martikainen, "Blocking of Dynamic Multicast Connections in a Single Link", in Proceedings of Broadband Communications'98, BC'98, Stuttgart, Germany, April 1998, pp. 473–483.

[18] J. Karvo, J. Virtamo, S. Aalto and O. Martikainen, "Blocking of Dynamic Multicast Connections", Telecommunication Systems, vol. 16, no. 3–4, 2001, pp. 467–481.

[19] J. S. Kaufman, "Blocking in a Shared Resource Environment", IEEE Transactions on Communications, vol. 29, no. 10, 1981, pp. 1474 – 1480.

[20] S. M. Kay, "Fundamentals of Statistical Signal Processing: Estimation Theory", Prentice-Hall, 1993.

[21] F. P. Kelly, "Blocking Probabilities in Large Circuit Switched Networks", Advances in Applied Probability, vol. 18, 1986, pp. 473–505.

[22] P. Lassila, J. Virtamo, "Efficient Monte Carlo Simulation of Product Form Systems", in Proceedings of the 14^{th} Nordic Teletraffic Seminar, NTS 14, Copenhagen, Denmark, August 1998, pp. 355–366.

[23] A. M. Law, W. D. Kelton, "Simulation Modeling & Analysis", McGraw-Hill International Editions, 1991, 759 p.

[24] G. Louth, M. Mitzenmacher, F. Kelly, "Computational Complexity of Loss Networks", Theoretical Computer Science 125, 1994, pp. 45–59.

[25] M. Mandjes, "Fast Simulation of Blocking Probabilities in Loss Networks", European Journal of Operations Research, Vol. 101, 1997, pp. 393-405.

[26] D. Mitra, J. A. Morrison and K. G. Ramakrishnan, "Unified Approach to Multirate ATM Network Design and Optimization", in Proceedings of the 9^{th} ITC Specialist Seminar on Teletraffic Modeling and Measurement in Broadband and Mobile Communications, Leidschendam, The Netherlands, November 1995, pp. 77–94.

[27] E. Nyberg, J. Virtamo and S. Aalto, "An Exact Algorithm for Calculating Blocking Probabilities in Multicast Networks", in Proceedings of Networking 2000, Paris, France, May 2000, pp. 275–286.

[28] S. Parekh, J. Walrand, "A Quick Simulation Method for Excessive Backlogs in Networks of Queues", IEEE Transactions on Automatic Control, vol. 34, no. 1, 1989, pp. 54–66.

[29] J. W. Roberts, "A Service System with Heterogenous User Requirements - Application to Multi-Services Telecommunications Systems", Performance of Data Communication Systems (G. Pujolle ed.), North-Holland Publishing Company, 1981, pp. 423–431.

[30] J. Roberts, U. Mocci, J. Virtamo (eds.), "Broadband Network Teletraffic – Final Report of Action COST 242", Springer, Lecture Notes in Computer Science, 1996, 584 p.

[31] K. W. Ross, "Multiservice Loss Models for Broadband Telecommunication Networks", Springer-Verlag, 1995, 343 p.

[32] R. Y. Rubinstein, B. Melamed, "Modern Simulation and Modeling", Wiley Series in Probability and Statistics, 1998.

[33] J. S. Sadowsky, J. A. Bucklew, "On Large Deviations Theory and Asymptotically Efficient Monte Carlo Estimation", IEEE Transactions on Information Theory, vol. 36, no. 3, 1990, pp. 579–588.

[34] J. S. Sadowsky, "Large Deviations Theory and Efficient Simulation of Excessive Backlogs in a GI/GI/m Queue", IEEE Transactions on Automatic Control, vol. 36, no. 12, 1991, pp. 1383–1394.

[35] A. Simonian, J. W. Roberts, F. Théberge, R. Mazumdar, "Asymptotic Estimates for Blocking Probabilities", Advances in Applied Probability, vol. 29, 1997, pp. 806–829.

[36] L. Tierney, "Markov Chains for Exploring Posterior Distributions", The Annals of Statistics, vol. 22, No. 4, 1994, pp. 69–100.

[37] M. Villén–Altamirano, J. Villén–Altamirano, "RESTART: A Method for Accelerating Rare Event Simulations", in Proceedings of the 13^{th} International Teletraffic Congress, ITC 13, Copenhagen, Denmark, June 1991, pp. 71–76.

Analysis of the RED Buffer Management Algorithm

[38] E. Altman, K. Avrachenkov, C. Barakat, "A Stochastic Model of TCP/IP with Stationary Random Losses", in Proceedings of ACM SIGCOMM 2000, Stockholm, Sweden, September 2000, pp. 231–242.

[39] E. Altman, K. Avrachenkov, C. Barakat, R. Núñez-Queija, "State Dependent M/G/1 Type Queuing Analysis for Congestion Control in Data Networks", in Proceedings of IEEE INFOCOM 2001, Anchorage, USA, April 2001, pp. 1350–1359.

[40] F. M. Anjum, L. Tassiulas, "Fair Bandwidth Sharing among Adaptive and Non-Adaptive Flows in th Internet", in Proceedings of IEEE INFOCOM 1999, New York, USA, March 1999, pp. 1412–1420.

[41] S. Athuraliya, S. Low, D. Lapsley, "Random Early Marking", in Proceedings of the 1^{st} COST 263 International Workshop: Quality of Future Internet Services, QofIS 2000, Berlin, Germany, September 2000, pp. 43–54.

[42] B. Braden et al., "Recommendations on Queue Management and Congestion Avoidance in the Internet", RFC2309, 1998.

[43] P. Brown, "Resource Sharing of TCP Connections with Different Round Trip Times", in Proceedings of IEEE INFOCOM 2000, Tel Aviv, Israel, March 2000, pp. 1734–1741.

[44] T. Bu, D. Towsley, "Fixed Point Approximations for TCP Behavior in an AQM Network", to be presented at ACM SIGMETRICS 2001, Cambridge, Massachusetts, USA, June 2001, report available from `ftp://www-net.cs.umass.edu/pub/Bu_fixedpoint_00.pdf`.

[45] D. Clark, W. Fang, "Explicit Allocation of Best-Effort Packet Delivery Service", IEEE/ACM Transactions of Networking, vol. 6, no. 4, 1998, pp. 362–373.

[46] O. Diekmann, S. A. van Gils, S. M. Verduyn Lunel, and H.-O. Walther, "Delay Equations: Functional-, Complex-, and Nonlinear Analysis", Applied Mathematical Sciences, Springer Verlag, 1995.

[47] W. Feng, D. D. Kandlur, D. Saha, K. G. Shin, "A Self-Configuring RED Gateway", in Proceedings of IEEE INFOCOM 1999, New York, USA, March 1999, pp. 1320–1328.

[48] W. Feng, D. D. Kandlur, D. Saha, K. G. Shin, "Stochastic Fair Blue: A Queue Management Algorithm for Enforcing Fairness", in Proceedings of IEEE INFOCOM 2001, Anchorage, Alaska, USA, April 2001, pp. 1520–1529.

[49] V. Firoiu, M. Borden, "A Study of Active Queue Management for Congestion Control", in Proceedings of IEEE INFOCOM 2000, Tel Aviv, Israel, March 2000, pp. 1435–1444.

[50] S. Floyd, V. Jacobson, "Random Early Detection Gateways in Congestion Avoidance", IEEE/ACM Transactions on Networking, vol. 1, no. 3, 1993, pp. 397–413.

[51] S. Floyd, "RED: Discussions of Setting Parameters", available from `http://www.aciri.org/floyd/REDparameters.txt`, 1997.

[52] S. Floyd, K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet", IEEE/ACM Transactions on Networking, vol. 7, no. 4, August 1999, pp. 458–472.

[53] S. Floyd, "Recommendation on using the "gentle_" variant of RED", available from `http://www.aciri.org/floyd/red/gentle.html`, 2000.

[54] D. Gross, C. Harris,"Fundamentals of Queueing Theory", Wiley, 3rd edition, 1998.

[55] J. K. Hale, "Introduction to Functional Differential Equations", Applied Mathematical Sciences, Springer Verlag, 1993.

[56] C. Hollot, V. Misra, W. Gong, D. Towsley,"A Control Theoretic Analysis of RED", in Proceedings of IEEE INFOCOM 2001, Anchorage, USA, April 2001, pp. 1510–1519.

[57] C. Hollot, V. Misra, W. Gong, D. Towsley,"On Designing Improved Controllers for AQM Routers Supporting TCP Flows", in Proceedings of IEEE INFOCOM 2001, Anchorage, USA, April 2001, pp. 1726–1734.

[58] F. P. Kelly, "Charging and Rate Control for Elastic Traffic", European Transactions on Telecommunications, vol. 8, no. 1, 1997, pp. 33-37.

[59] F. Kelly, "Mathematical Modeling of the Internet", available from `http://www.statslab.cam.ac.uk./~frank/` (also an extended version in Mathematics Unlimited – 2001 and Beyond, Springer Verlag, 2000), 1999.

[60] P. Kuusela, J. T. Virtamo, "Modeling RED with Two Traffic Classes", in Proceedings of the 15^{th} Nordic Teletraffic Seminar, NTS 15, Lund, Sweden, August 2000, pp. 271–282, available from `http://www.tct.hut.fi/tutkimus/com2/publ/red2cl.pdf`.

[61] S. Köhler, M. Menth, N. Vicari, "Analytic Performance Evaluation of the RED Algorithm for QoS in TCP/IP Networks", to be presented at the 2^{nd} COST 263 International Workshop: Quality of Future Internet Services, QofIS 2001, Coimbra, Portugal, September 2001, report available from `http://www-info3.informatik.uni-wuerzburg.de/staff/koehler/paper/tr259.pdf`.

[62] D. Lin, R. Morris, "Dynamics of Random Early Detection", in Proceedings of ACM SIGCOMM 1997, Sophia Antipolis, France, September 1997, pp. 127-137.

[63] M. Mathis, J. Semke, J. Mahdavi, T. Ott, "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm", ACM Computer Communications Review, vol. 27, no. 3, July 1997, pp. 67–82.

[64] M. May, T. Bonald, J. Bolot, "Analytic Evaluation of RED Performance", in Proceedings of IEEE INFOCOM 2000, Tel Aviv, Israel, March 2000, pp. 1415–1424.

[65] A. Misra, T. J. Ott, "The Window Distribution of Idealized TCP Congestion Avoidance with Variable Packet Loss", in Proceedings of IEEE INFOCOM 1999, New York, USA, March 1999, pp. 1564–1572.

[66] V. Misra, W. Gong, D. Towsley,"A Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED", in Proceedings of ACM SIGCOMM 2000, Stockholm, Sweden, September 2000, pp. 151–160.

[67] T. J. Ott, T. V. Lakshman, L. H. Wong, "SRED: Stabilized RED", in Proceedings of IEEE INFOCOM 1999, New York, USA, March 1999, pp. 1346–1355.

[68] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, "Modeling TCP Throughput: A Simple Model and its Empirical Validation", in Proceedings of ACM SIGCOMM 1998, Vancouver, Canada, September 1998, pp. 303–314.

[69] S. Peeters, C. Blondia, "A Discrete Time Analysis of Random Early Detection with Responsive Best-Effort Traffic", in Proceedings of the 7^{th} IFIP Working Conference on Performance Modeling and Evaluation of ATM & IP Networks, IFIP ATM & IP 1999, Antwerp, Belgium, June 1999.

[70] L. L. Peterson, B. S. Davie, "Computer Networks - A Systems Approach (2nd ed.)", Morgan Kaufman Publishers Inc., 1999, 776 p.

[71] V. Sharma, J. Virtamo, P. Lassila, "Performance Analysis of the Random Early Detection Algorithm", submitted for publication, available from `http://www.tct.hut.fi/tutkimus/com2/publ/redanalysis.pdf`, 2000.

[72] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery Algorithms", RFC2001, 1997.

[73] D. K. H. Tan, "Mathematical Models of Rate Control for Communication Networks", Ph.D. Thesis, University of Cambridge, August 1999.

[74] H. C. Tijms, "Stochastic Models: An Algorithmic Approach", Wiley, 1995.

[75] W. Wang, D. Tipper, S. Banerjee, "A Simple Approximation for Modeling Nonstationary Queues", in Proceedings of IEEE INFOCOM 1996, San Fransisco, USA, March 1996, pp. 255–262.

[76] T. Ziegler, S. Fdida, U. Hofmann, "RED+ Gateways for Identification and Discrimination of Unfriendly Best-Effort Flows in the Internet", in Proceedings of the 7^{th} IFIP Working Conference on Performance Modeling and Evaluation of ATM & IP Networks, IFIP ATM & IP 1999, Antwerp, Belgium, June 1999, pp. 27–38.

Errata

Publication 2

- General: The sets $\mathcal{A}_i, \mathcal{B}, \mathcal{S}$ should appear in a calligraphic font, but apparently IEE Electronics Letters does not use these fonts. Instead, all sets have been typeset by using a normal bold faced font.
- p. 1204, first sentence in the paragraph "Variance Reduction": the set S in $X \in S$ should read $X \in \mathbf{S}$.
- p. 1205, before eq. (5): ρk should appear ρ_k

Publication 3

- last part of eq. (2), the expectation should have $\mathbf{X}$ in the subscript (signifying a random variable), instead of $\mathbf{x}$

Publication 4

- p. 329, equation for $\pi(\mathbf{x})$: in the ratio, the X should be $\mathbf{X}$ denoting a vector random variable
- p. 335, below Fig. 2: the equation in the text $\mathrm{P}\{Y_l \leq y \,|\, S_l = x \geq U\}$ should read $\mathrm{P}\{Y_l \leq y \,|\, S_l = x\} \geq U$
- p. 335, text explaining Fig. 2: the sentence starting "The most important ..." contains an extra "S_l^j or", that is, in the middle of the sentence, the text should read "... for a fixed value of S_l are ..."
- p. 336, equation for $p^*(\mathbf{x})$: in the ratio, the X should be $\mathbf{X}$ denoting a vector random variable; the same mistake appears in the equation for v below
- p. 337, last equations on the page: in the equations on the right hand side for z_i, the subscript of z_k is false, i.e., z_k should be z_K
- p. 338: an '='-sign is missing from eq. (11) (it should read $\tilde{\mathbf{\Gamma}}^{-1} = ...$)
- p. 340, equations after definition of $p^*(\mathbf{x})$: there are some mix-ups in the subscripts of the variables for r' and s'. The correct version reads: "... strip $v = Q(r') - Q(s')$, the primes refer to the normalized variables

$$x'_k = \frac{x_k - m_k}{\sigma_k}, \quad r' = \frac{r - \widetilde{m}_K}{\sqrt{\widetilde{\Gamma}_{KK}}}, \quad s' = \frac{s - \widetilde{m}_K}{\sqrt{\widetilde{\Gamma}_{KK}}}, \ldots"$$

- p. 341, sec. 6.1, line 4: $\beta = \sum_j \hat{\eta}^j$ should be $\beta = \sum_j \eta^j$
- p. 341, first equation in Section 6: $\hat{\eta}^{(j)} = \ldots$ should be $\hat{\eta}^j = \ldots$
- p. 341, sec. 6.1, line 11: "estimators $\hat{\beta}_j$ are ..." should be "estimators $\hat{\eta}^j$ are ..."
- p. 342, first paragraph of Section 6.2: The definition of relative deviation in the sentence "To this end, we estimated the relative deviation of the estimator..." should have brackets around $\hat{\beta}_k$, i.e., $(\mathrm{V}[\hat{\beta}_k])^{1/2}/\hat{\beta}_k$

Publication 5

- p. 435, left column, equation for the IS estimator: In the text after the equation, v_j should be v_u^j
- p. 436, equation before eq. (8): v_j should read v_u^j (similarly in the sentence preceding the equation)

Publication 6

- Section 3: The error terms in the equations appearing between eqs. (2) and (3) should be $O((\Delta t)^2)$, instead of $O(\Delta t)$
- Section 3: The justification of the ODE for $\bar{q}(t)$ contains some inaccuracies and a corrected version is given in the main text of this thesis, see Section 3.4.3
- Section 5.1: text above Fig. 2: The list of the RED parameters that were used to obtain the figures is not correct. The correct parameter values should be: $K = 100$, $T_{\min} = 40$, $T_{\max} = 80$, and $p_{\max} = 0.2$.