Helsinki University of Technology Networking Laboratory
Teknillinen korkeakoulu Tietoverkkolaboratorio
Espoo 2002 Report 6/2002

A STUDY OF TELETRAFFIC PROBLEMS IN MULTICAST NETWORKS

Jouni Karvo

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 29th of November, 2002, 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

ISBN 951-22-6121-9
ISSN 1458-0322

Picaset Oy
Espoo 2002

TEKNILLINEN KORKEAKOULU
TEKNISKA HÖGSKOLAN
HELSINKI UNIVERSITY OF TECHNOLOGY

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

Author: Jouni Karvo

Name of the dissertation
A Study of Teletraffic Problems in Multicast Networks

Date of manuscript	Oct. 1, 2002	Date of the dissertation	Nov. 29th, 2002
☐ Monograph		☒ Article dissertation (summary + original articles)	

Department	Department of Electrical and Communications Engineering
Laboratory	Networking Laboratory
Field of Research	teletraffic theory, performance evaluation of communications networks
Opponent	Professor Villy Bæk Iversen (Technical University of Denmark)
Supervisor	Professor Jorma Virtamo (Helsinki University of Technology)
Instructor	Prof. Jorma Virtamo, Dr. Samuli Aalto

Abstract

This dissertation studies teletraffic engineering of dynamic multicast connections. The traditional models in teletraffic engineering do not handle multicast connections properly, since in a dynamic multicast tree, users may join and leave the connection freely, and thus the multicast tree evolves in time.

A model called multicast loss system is used to calculate blocking probabilities in a single link and in tree-type networks. In a single link case, the problem is a generalised Engset problem, and a method for calculating call blocking probabilities for users is presented. Application of the reduced load approximation for multicast connections is studied. Blocking probabilities in a cellular system are studied by means of simulation.

The analysis is mainly concentrated on tree type networks, where convolution-truncation algorithms and simulation methods for solving the blocking probabilities exactly are derived. Both single layer and hierarchically coded streams are treated. The presented algorithms reduce significantly the computational complexity of the problem, compared to direct calculation from the system state space. An approximative method is given for background traffic.

The simulation method presented is an application of the Inverse Convolution Monte-Carlo method, and it gives a considerable variance reduction, and thus allows simulation with smaller sample sizes than with traditional simulation methods.

Signalling load for dynamic multicast connections in a node depends on the shape of the tree as well as the location of the node in the tree. This dissertation presents a method for calculating the portion of signalling load that is caused by call establishments and tear-downs.

Keywords	multicast, blocking probability, loss systems, signalling load, Monte-Carlo simulation, convolution		
UDC	621.395:519.2:004.7	Number of pages	72 p.
ISBN (printed)	951-22-6121-9	ISBN (pdf)	951-22-6122-7
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 dissertation is the result of the work started originally in the TOVE project in 1997. The project was directed by professor Olli Martikainen in the Telecommunications Software and Multimedia Laboratory in the Helsinki University of Technology. We noticed that there was no teletraffic theory for dimensioning multicast networks, for example for TV or radio delivery in broadband networks.

Already in the beginning, professor Jorma Virtamo and doctor Samuli Aalto started co-operation with us. After I finished my licentiate's thesis, professor Virtamo took the official responsibility of supervising this dissertation.

Starting from August, 2001, I work with the Laboratory of Telecommunications Technology (current Networking Laboratory), first in the COM^2 project and later in the COST279 project. Nokia Foundation and Tekniikan edistämissäätiö have also funded my work in form of grants.

Recently, the networking technology and markets have matured so that field trials for TV delivery via a broadband network have started. Helsingin Sanomat reported on Saturday 26th of January 2002 that Elisa Communications and Maxisat have started a trial in eastern Helsinki. This implies that the time has come for possible real-life applications of the algorithms and models in this dissertation.

I thank Prof. Jorma Virtamo and Dr. Samuli Aalto for their endless patience in instructing me during this work, and Prof. Olli Martikainen for originally giving me this subject. I also thank Dr. Pasi Lassila and Mr. Janne Aaltonen for their work as co-authors in the publications. I thank the pre-examiners, Prof. I. Norros and Prof. M. Pióro, for giving a wealth of comments for improving the presentation of the dissertation. The laboratory staff at the Laboratory of Telecommunications Software, and at the Networking Laboratory have both been important for creating a good working atmosphere. Finally, I thank my wife Johanna for her support.

Espoo, on the 1st of October 2002

Jouni Karvo

CONTENTS

LIST OF PUBLICATIONS

[1] KARVO, J., VIRTAMO, J., AALTO, S., AND MARTIKAINEN, O. Blocking of dynamic multicast connections in a single link. In *Proc. of International Broadband Communications Conference, Future of Telecommunications* (Stuttgart, Germany, Apr. 1998), pp. 473–483.

[2] KARVO, J., MARTIKAINEN, O., VIRTAMO, J., AND AALTO, S. Blocking of dynamic multicast connections. *Telecommunication Systems 16*, 3,4 (2001), 467–481.

[3] KARVO, J., AND AALTO, S. Average signalling load for multicast group management. In *Proc. International Teletraffic Congress ITC-16* (Edinburgh, Great Britain, June 1999), D. Smith and P. Key, Eds., pp. 509–518.

[4] AALTO, S., KARVO, J., AND VIRTAMO, J. Calculating blocking probabilities in multicast loss systems. In *Proc. Intl Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2002)* (San Diego, CA, July 2002), pp. 833–842.

[5] KARVO, J., AALTO, S., AND VIRTAMO, J. Blocking probabilities of two-layer statistically indistinguishable multicast streams. In *Proc. International Teletraffic Congress ITC-17* (Salvador da Bahia, Brazil, Sept. 2001), J. M. de Souza, N. L. S. Fonseca, and E. A. de Souza e Silva, Eds., pp. 769–779.

[6] KARVO, J., AALTO, S., AND VIRTAMO, J. Blocking probabilities of multi-layer multicast streams. In *2002 Workshop on High Performance Switching and Routing (HPSR 2002)* (Kobe, Japan, May 2002), pp. 268–277.

[7] LASSILA, P., KARVO, J., AND VIRTAMO, J. Efficient importance sampling for Monte Carlo simulation of multicast networks. In *Proc. INFOCOM'01* (Anchorage, Alaska, Apr. 2001), pp. 432–439.

[8] KARVO, J. Efficient simulation of blocking probabilities for multi-layer multicast streams. In *Proc. IFIP Networking 2002* (Pisa, Italy, May 2002), E. Gregori, M. Conti, A. T. Campbell, G. Omidyar, and M. Zukerman, Eds., vol. 2345 of *Lecture Notes in Computer Science (LNCS)*, pp. 1020–1031.

[9] AALTONEN, J., KARVO, J., AND AALTO, S. Multicasting vs. unicasting in mobile communication systems. In *Proc. Workshop on Wireless Mobile Multimedia, WoWMoM 2002* (Atlanta, GA, Sept. 2002), pp. 104–108.

LIST OF SYMBOLS

These symbols are used in the introductory part of the dissertation. Notation differs in the publications.

B	blocking probability
$\mathcal{B}$	set of blocking states
c_i	capacity requirement of channel i
$\mathbf{c}$	capacity requirement vector
C_j	capacity of link j
$d(\cdot)$	capacity requirement function
I	number of multicast channels, index of the channel for which blocking probability is calculated
$\mathcal{I}$	set of multicast channels
J	number of links
$\mathcal{J}$	set of links
L	number of layers
$\mathcal{L}$	set of layers
$\mathcal{M}_j$	set of links downstream of link j (including link j)
$\mathcal{N}_j$	set of downstream neighbour links of link j
Q	occupancy vector
$\mathcal{R}_u$	route from link u to the root of the tree (set of links)
$\mathcal{R}_j^u$	part of the route from link u to the root of the tree that is downstream of link j
$\mathcal{S}$	state space of a link
$\mathcal{U}$	set of user links
$\mathcal{U}_j$	set of user links downstream of link j
$Y_{j,i}$	the state random variable of channel i in link j
$\mathbf{X}$	the network state random variable
$\mathbf{Y}_j$	the state random variable of link j
$\pi(\mathbf{x})$	probability of state $\mathbf{x}$
$\tilde{\pi}(\mathbf{x})$	probability of state $\mathbf{x}$, with condition $\mathbf{x}$ in the allowed state space

$\pi_j(\mathbf{y})$ probability of state $\mathbf{y}$ in link j

$\pi_j(\phi)$ probability that link j is in one of the states in ϕ

ϕ partial state descriptor

φ partial state descriptor for channels $i \in \mathcal{I} - \{I\}$

$\tilde{\Phi}_j$ allowed partial state descriptor space for link j

$\tilde{\Phi}_{j,r}$ allowed partial state descriptor space for link j and request of capacity r

Ω network state space

$\tilde{\Omega}$ the set of allowed network states

1 INTRODUCTION

Teletraffic theory is a field of science emanating from the pioneering work of Agner Krarup Erlang (1878–1929) and Tore Olaus Engset (1865–1943), in the first half of the 20th century. They created formulae that can be used for calculating the required capacity for a telephone trunk group to carry the offered telephone calls. The work of Erlang and Engset was based on theory of probability, which was being developed during the same epoch.

The basic problem is characterised by the concepts of quality or grade of service, capacity, and traffic.

- The traditional measure used in conjunction with the grade of service is the call blocking probability—the probability that a user cannot establish a call. Call blocking might happen due to the other user being busy (subscriber line blocking) or due to insufficient capacity in the network.

- Capacity is traditionally measured in channels. The old analog systems reserved a “wire” or sub-carrier for each call. When the telephone system was digitalised in the 1970’s, the calls were given constant 64 kbit/s (or 56 kbit/s) PCM channels. The ISDN in the 1980’s, and the subsequent ATM changed the scene by allowing different calls to have different capacity requirements.

- Traffic is generated by the users of the telephone system. When a user establishes a call, speaks, and then terminates the call, he contributes to the overall traffic process of the telephone system. The traditional Erlang model assumes that all users are independent human beings, and that they are numerous. The duration of the users’ calls is called *call holding time*. The traditional measure for offered traffic is offered traffic intensity, which gives the mean number of active telephone calls on a hypothetical infinite capacity link. Traffic intensity is dimensionless, but C.C.I.F. (Comité Consultatif International Téléphonique, a predecessor of ITU-T and C.C.I.T.T.) decided in 1946 that `Erlang` is used as the traffic unit to honour A. K. Erlang’s work.

Traffic generated by the users of the network is sometimes more voluminous than at other times. This variation is natural; typically telephones have been used more in business than in leasure. The networks need to be dimensioned so that even during the most hectic times, the grade of service is acceptable. For this use, a concept called *busy hour* was defined. There are several definitions of busy hour, but their intention is the same: to capture the peak traffic time.

The data transmitted in a digital network needs to be coded, i.e. the signal is sent as a series of numbers. There are many different coding standards, such as the PCM coding for telephone traffic, and MPEG coding for video streams. Audio and video streams can be coded hierarchically, where information is separated according to its importance, and then coded and transmitted in separate streams.

In addition to the actual traffic in the network, the network has to transport control information between the network elements. This is called signalling. The early telephone network used in-band signalling; the users requested from the operator to be connected to a specific number. The modern telecommunications networks offer signalling protocols for the network elements and applications to use. The telephone network uses the ITU-T common channel signalling system version 7, where a dedicated network carries signalling information. This signalling network may either have dedicated links or use the capacity of the controlled network. The Internet does not separate signalling from data traffic.

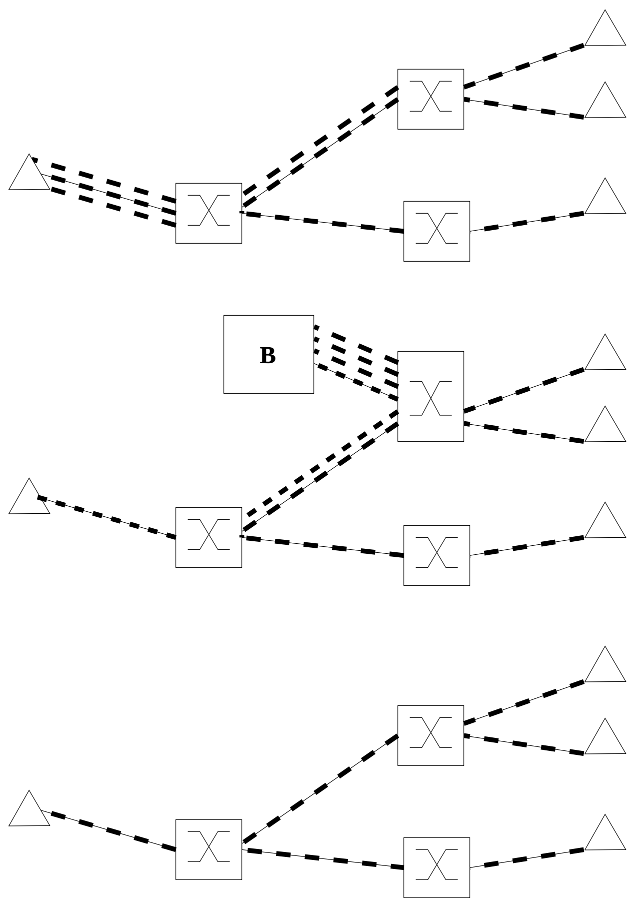

Figure 1.1: Capacity usage and unicast vs. multicast connections. Top: unicast connections made by the sender, middle: bridge B copies the stream of the sender to the other participants, bottom: native multicast, where network elements make the required copies of the the stream. Data streams are presented with a thick broken line.

Multicast is a technique where there is one sender and multiple receivers for the same transmission. Videoconferencing is a classical application of multicast. In the traditional network, multicast connections are not supported, but a separate server called “bridge” is used. Some more recent network devices allow the network itself to provide multicast.

Multicast connections characteristically economise transmission capacity. This means that a multicast connection — in taking the form of a tree, where streams branch at the nodes — requires much less capacity from the network links than a bunch of separate point to point (unicast) connections from the root node to the leaf nodes of the tree (see Figure 1.1). This effect poses a problem when dimensioning a network. With multicast trees, less bandwidth is needed near the root of the tree than the sum of the bandwidths used near the leaves.

The applicability of the traditional methods for dimensioning telecommunications networks to multicast is limited. In the case where multicast connections are static, that is, the multicast connection to all users of the session is established simultaneously (and torn down simultaneously), the existing theory applies; the set of network resources needed by each connection is reserved during connection establishment. In the case where users of the multicast connection can freely join and leave the multicast tree, a new theory is needed. This dissertation treats this problem as a continuation of the classical teletraffic theory.

Even though most of the results in this dissertation are algorithms giving exact results, their application is subject to approximative assumptions. Here is a list of factors that affect the accuracy of the solutions:

- Traffic intensities. For a dimensioning task, traffic intensities need to be estimated. Due to the approximate nature of the input values, the results are approximative.

- Background traffic. The models presented in this dissertation are first derived for multicast traffic in isolation. Inclusion of unicast traffic using the concept of “independent background traffic” leads to an approximation, since the background traffic in general is not independent between links. When dimensioning, summing the resulting capacity demands for links for both unicast and multicast traffic leads to a conservative estimate, given the traffic process assumptions are valid, due to statistical multiplexing.

- Traffic process. The models used mostly assume Poisson arrivals but general holding times for calls. Where some other assumptions hold, it is mentioned. As a call-scale model, Poisson arrivals reflect behaviour of independent human users well. The presence of automated equipment, such as automatic overnight back-up systems, or modems, reduces the accuracy of the assumption.

- Link and connection capacities. The models use integers for link capacities and connection capacity demands. This is not a constraint when only constant bit-rate connections appear in the system, due to the nature of digital systems. When variable bit-rate connections, however, appear, the resulting efficient bandwiths themselves are approximations and furthermore, often cannot be presented as integer multiples. In both cases, the accuracy of calculations can be made sufficient by making the capacity unit small enough.

This dissertation presents new algorithms for calculating blocking probabilities for multicast traffic. Multicast connections are analysed in a single link, as well as in tree-type networks. Single-layer and hierarchically coded streams are treated, and convolution and simulation approaches are developed for calculating blocking probabilities. A method for calculation of signalling load generated by users joining and leaving the multicast trees is introduced.

The dissertation is organised as follows. Chapter 2 gives a summary of the techniques and background of the study. Then, Chapter 3 presents the system model and notation. A method for calculating blocking probabilities in a single link, and an approximative method for calculating end-to-end blocking probabilities by using the single link model is given in Chapter 4. The chapter also includes a study of simulation of multicast blocking probabilities in a cellular system with a large number of possible multicast trees. Chapter 5 presents new convolution truncation algorithms for tree-type networks. The algorithms give a considerable reduction in the computational effort required when compared to calculating directly the probabilities of the possible network states. The algorithms can be used in systems where multicast connections are in isolation, but they can also be adapted to the case where there is background traffic present, in which case they give approximative results. Chapter 5 also presents a new algorithm for fast simulation of multicast blocking probabilities. Chapter 6 shows how signalling load for multicast group management can be calculated, and finally, Chapter 7 presents a summary of the work, and outlines the contributions of this dissertation. Appendix I presents a proof omitted from Publication [6] due to space restrictions.

2 LOSS SYSTEMS

A loss system is a mathematical model where a set of users (or classes) compete of a finite set of resources. The users generate requests according to a stochastic process. If the resources required are available, the requests are admitted and the user holds the requested resources for a random period, called *holding time*, and then releases them. If the requested resources are not available at the time of request, the request is lost. There are no queues or re-trials in the model. In the case of a lost request, the user continues as if the resource had just been freed. Typically, additional assumptions are made in order to be able to describe the behaviour of the loss system as a Markov processes.

A typical system modelled as a loss system is the telephone network, where the capacity in the links is the resource competed for, requested by calling telephone users. The classical example used in telephone network dimensioning is the Erlang formula for blocking probabilities, or the Erlang-B formula:

$$B_c = \frac{\frac{a^C}{C!}}{\sum_{i=0}^{C} \frac{a^i}{i!}},$$

where a denotes the offered traffic intensity during the busy hour (Poisson process), C the link capacity (in channels), and B_c the call blocking probability.

2.1 Reversibility

A stochastic process is called reversible if it remains the same when the direction of time is reversed. In general, the resulting process in the reverse time is not the same as in normal time. A Markov process is reversible if and only if it satisfies the *detailed balance conditions*, i.e. for all pairs of states (j, k) it holds:

$$\pi(j)q(j,k) = \pi(k)q(k,j),$$

where $q(j,k)$ and $q(k,j)$ denote the transition rates from state j to k and from state k to j, respectively [46]. That is, the probability flows between each pair of states are equal. Reversibility was first studied by Kolmogorov [48], who established that detailed balance is a necessary and sufficient condition for the transition probabilities in the original process and in the reversed process to be identical. Later, Kelly [46] developed applications for reversibility.

2.2 Truncation principle

Erlang's formula is a good example of an application of the so-called truncation principle [46, page 25]; a technique used extensively in this dissertation. The system behind the formula is the $M/M/n/n$-server where $n = C$

is the number of servers (channels). The system has $C+1$ states. Now, consider a $M/M/\infty$-system. The probability $\pi(i)$ of state i in this system is

$$\pi(i) = \frac{a^i}{i!}e^{-a}. \tag{2.1}$$

According to the truncation principle, a reversible Markov process can be altered as follows: Partition the states of the system to two sets (A, B). Then, scale all the intensities of transitions from set A to set B with a constant c. The equilibrium distributions of the sets will be

$$G^{-1}\pi(j), \text{ for } j \in \mathsf{A}, \qquad \text{and} \quad G^{-1}c\pi(j), \text{ for } j \in \mathsf{B}.$$

Defining A to the set of states i where $i \leq C$ and setting $c := 0$, the resulting system is the $M/M/n/n$-server. The only remaining thing is calculating the normalisation constant G, which is (in this case) the sum of the state probabilities in the set A.

The truncation principle applied to loss systems states that the system state probability distribution can be calculated first as if there were no capacity restrictions (in an infinite system), and then normalising the probabilities. Let Ω denote the state space of the (infinite) system, and $\tilde{\Omega}$ the state space of the system with capacity constraints. Consider a set of blocking states $\mathcal{B} \subseteq \tilde{\Omega}$, such as the set of states where no extra calls are allowed. The probability B that the system is in one of the blocking states is then

$$B = \frac{\mathrm{P}\{\, \mathbf{X} \in \mathcal{B} \,\}}{\mathrm{P}\{\, \mathbf{X} \in \tilde{\Omega} \,\}}. \tag{2.2}$$

This equation forms the basis of the blocking probability calculation. It is often quoted that the most difficult or time consuming task in calculating blocking probabilities is calculating $G = \mathrm{P}\{\, \mathbf{X} \in \tilde{\Omega} \,\}$, due to the often large number of states in the set $\tilde{\Omega}$.

The loss systems are reversible in the case where there are no capacity constraints. For the case with capacity constraints, Aein [12] derived a sufficient condition for the system to have a truncated product form distribution, called co-ordinate convexity. In general, reversibility is a sufficient condition for application of the truncation principle to Markov processes [46], but there are some other systems where it can be used, too [23].

2.3 Insensitivity

If the stationary distribution of a loss system remains the same even if the holding time distribution is not exponential, the stationary distribution is said to be insensitive with regard to the holding time distribution. Insensitivity in the case of the Erlang B-formula was already studied in the original 1917 paper [36]; Erlang started with constant holding times and then studied the exponential case. Palm [60] showed that the Erlang system is indeed insensitive with respect to holding times. The Engset's original paper [34] assumed that users are independent of each other, but did not make any assumptions about holding time distribution. Cohen [28] showed the insensitivity result for the holding times.

According to Whittle [74], König and Matthes studied insensitivity in a more general setting. Later, Whittle [73] used imbedded Markov chains to derive a result of insensitivity using partial balance. Schassberger [69] derived an insensitivity result for Markov chains with countable sets of states, restricted to convolutions of exponential distributions. Whitt [72] derived a convergence theorem for finite-state continuous time generalised semi-Markov processes, showing the phase method used by Schassberger to be applicable to any distribution. Later, Schassberger [70] extended the result for $M/G/\infty$ type systems.

Schassberger's insensitivity framework associates the system with a countable set S of clocks. Each state j of the system is associated with set S_j denoting the set of clocks active in state j. The value of each active clock is reducing at a non-negative rate $c(s,j)$, and when the first active clock reaches the value of zero, a state transition is triggered. In the state transition, the values of the clocks that are active in both states (except the triggering clock) are unaffected, but the other clocks are given new values, drawn from a given lifetime distribution with mean λ_s^{-1}. The insensitivity result states that if there is local balance with respect to a set of clocks, then the system is insensitive with respect to the corresponding lifetime distribution. The local balance equations are:

$$\begin{aligned}\pi(j)\lambda_s c(s,j) = \sum_{k\in\Omega-\Omega_s} \pi(k) \sum_{s'\in S_k} \lambda_{s'} c(s',k)p(k,s',j) \\ + \sum_{k\in\Omega_s} \pi(k)\lambda_s c(s,k)p(k,s,j),\end{aligned}$$

where $j \in \Omega_s$. Here, S_j denotes the set of clocks active in state j, $c(s,j)$ the rate at which clock s is running in state j, $\Omega_s = \{j : s \in S_j\}$, and $p(k,s,j)$ denotes the probability that the system jumps from state k to state j when clock s runs out. Now, extending the system state space by the states of the users, and using relabelling, the same result can be used for $M/G/\infty$ type systems [70].

2.4 Different types of blocking probabilities

There are different types of blocking probabilities that can be defined, namely:

- *Time blocking probability*, (*time congestion*) which is the probability that at an arbitrary instant of time there is no space in the system for any more calls of traffic class k, i.e. that the system is "full".
- *Call blocking probability*, (*call congestion*) which is the blocking probability experienced by a user, such as the ratio of unsuccessful call attempts to the total number of call attempts.

There is a close relationship between the time blocking probability and the call blocking probability. Assuming a system with infinite capacities and independent users, the system is of product form, i.e. the steady state distribution of the system is the product of its marginal distributions. Additionally, all user processes are assumed to have the detailed balance property. Then, the whole system has the detailed balance property, as well. Applying capacity constraints by using truncation preserves the product form of the system. The probability of system state $\mathbf{x} \in \tilde{\Omega}$ is then $\mathrm{P}\{\mathbf{X} = \mathbf{x}\} = G^{-1} \prod_{u' \in \mathcal{U}} \mathrm{P}\{X_{u'} = x_{u'}\}$, where $\mathcal{U}$ denotes the set of users of the system and $X_{u'}$ denotes the state of user u'. Consider the states $\mathbf{x}$ where user u has no active call, i.e. $x_u = 0$. If the user tries to make a call at an arbitrary moment (the call interarrival time is exponentially distributed), the call blocking probability B^c, can be calculated as

$$B^c = \mathrm{P}\{\mathbf{X} \in \mathcal{B} | X_u = 0\} = \frac{\mathrm{P}\{\mathbf{X} \in \mathcal{B}, X_u = 0\}}{\mathrm{P}\{X_u = 0\}} = \mathrm{P}\{\mathbf{X}' \in \mathcal{B}'\} = B'^t,$$

where $\mathbf{X}'$ and $\mathcal{B}'$ denote the system state and set of blocking states (where there is no room for a user u call) in a system without user u. The result, B'^t is the time blocking probability in a system without user u. That is, due to the relationship between time and call blocking probabilities, the call blocking probability can be calculated in the same way as the time blocking probability. The only difference needed is removing the user for which call blocking probability is calculated from the system.

As a special case, consider a Poisson arrival stream of calls. This arrival stream can be thought of as being generated by an infinite number of independent users. Approximate the blocking probability calculation by choosing an arbitrary finite number of users. Then calculate call and time blocking probabilities for this set of users. As the number of users increases (while keeping the total call arrival intensity constant), the difference between time and call blocking probabilities decreases, and in the limit, the time and call blocking probabilities are the same. This property follows from the PASTA (Poisson Arrivals See Time Averages) [75] property. The Erlang-B formula gives call blocking probabilities because of its Poisson arrival assumption.

Engset noticed the difference between time blocking probability and call blocking probability in his 1915 report [35]. Ross [64, page 162] gives the call blocking probability for multiservice loss models with finite user populations. The probability distributions seen by an arriving customer have been studied in queueing networks under the name *arrival theorem*, see e.g. [19].

2.5 Multiservice loss system

The Erlang formula is not applicable for connections with differing capacity demands. Dartois [29] studied the case of unbalanced traffic sources with equivalent capacity in a link, and derived a product form solution for the problem. Fortet and Grandjean [38] addressed this problem, and created an algorithm for calculating blocking probabilities in a single link in this case. Later, Kaufman [45] and Roberts [63] addressed the same problem. These algorithms had numerical problems in some cases. Nilsson *et al.* [55] created a new recursive algorithm to overcome these problems. Kuczura and Bajaj [49], Manfield and Downs [53], and Delbrouck [30] studied blocking probabilities in a link with Poisson streams and (non-Poisson) overflow traffic, as a continuation of the work of Wilkinson on Equivalent Random Theory (1956). Delbrouck [31] combined the results for overflow traffic to the recursive algorithm of Kaufman [45] and Roberts [63], allowing different traffic classes to have differing peakedness factors.

For calculating blocking probabilities exactly in the network case, the so-called multiservice loss system was developed. The multiservice loss system has a state space where each traffic class represents one dimension, having a traffic class specific marginal distribution that is the same as in the one traffic class case (Equation (2.1)). The traffic classes are independent before truncation, and thus the state probabilities are of product form. Product form systems were already treated by Jensen [41]. The truncation principle applies. The greatest difficulty in treating the multiservice loss networks is the possibly large number of states, which leads to difficulties in calculating the state sums, especially the denominator of Equation (2.2). For discussion on multiservice loss systems, see e.g. Ross' book [64].

For tree structured networks, Iversen [40] created a convolution algorithm, which was later refined by Tsang and Ross [71] for calculation by Fast Fourier Transform. Tsang and Ross [71] also generalised the recursive algorithm used by Kaufman and Roberts to the tree network case. Choudhury *et al.* [26, 25] present an algorithm for calculating blocking probabilities by numerically inversing generating functions. Lassila and Virtamo [51] provided efficient simulation methods for multiservice loss models.

The multiservice loss system does not allow modelling dynamic multicast connections, since the multiservice loss system requires the capacity for a call to be reserved simultaneously on all links.

2.6 Reduced Load Approximation

The well known Reduced Load Approximation (RLA) (see e.g. [64]) can be used to approximate end-to-end blocking probabilities. According to Ross [64], the earliest account of RLA is by R. B. Cooper and S. Katz (1964), and it was later studied further by Whitt and Kelly.

The idea of the RLA is that traffic blocked in a link on its route has little effect on the other links. In the multiservice loss model, the approximation yields fairly good results when no traffic class dominates and the total traffic intensity is not too high. Under these conditions, traffic in different links is nearly independent and Poissonian.

RLA consists of two alternating steps. First, assuming the call blocking probabilities for all traffic classes and links are known, it is possible to calculate the traffic intensity for traffic class i in link j. This is done by summing all offered traffic intensities for that traffic class after first thinning with the call blocking associated with them in each link along the route, with the exception of the link we are calculating blocking for. Second, knowing the traffic intensities for each traffic class i in link j, it is possible to calculate call blocking for each traffic class in the link.

The process is iterated, starting with for example zero blocking probability for each traffic class in each link, to give the approximate blocking probabilities.

In the case of single traffic class unicast traffic, RLA has a unique solution which is often accurate enough. This is due to Erlang blocking formula being strictly increasing, leading to situation where the fixed point equation can be formulated as a convex optimisation problem. There are several approximations for RLA in the multiservice case; Kelly's approximation (see [27]), and the Knapsack and Pascal approximations [27] which are more accurate than Kelly's approximation but may have several solutions. The loss of uniqueness of the solution is due to the blocking probabilities in the multiservice case not being strictly increasing (see Gimpelson [39]).

The Reduced Load Approximation, as explained in this section, does not take overflow traffic or alternative routing into account. Lin *et al.* [52] analysed a network with alternative routing approximating the overflow traffic (which in general is not a Poisson process) using the Poisson assumption. Dziong [33] took the dependence of traffic between different links into account by calculating the peakedness factor of the traffic on the path. Pióro *et al.* [61] have an early account on applying the single link recursion formula of Fortet, Grandjean, Kaufman and Roberts to evaluating the improvement of the grade of service when using alternative routing. The mentioned algorithms assume complete sharing as the link admission policy. Pióro *et al.* [62] studied algorithms for different policies.

2.7 Multicast

A lot of work has been done on multicast. It is not useful to try to give a complete review here, see instead Diot *et al.* [32] for a comprehensive treatise. After the publication of their paper, practically every conference and journal has continued to regularly publish articles on multicast. There are only a handful of papers, however, that deal with blocking probabilities and multicast.

Chan and Geraniotis [22] studied multicasting and subband coding (or hierarchical coding) of video. They define a system model, and use the Reduced Load Approximation for solving the blocking probabilities. They also noted that the traditional methods for multiservice loss systems do not apply in this case.

Our first paper, Publication [1] presents a model for calculating blocking probabilities of multicast connections in a single link. The obvious extension, application of Reduced Load Approximation followed in Publication [2]. Nyberg *et al.* [58] created the first convolution truncation algorithm for multicast. Boussetta and Belyot [20] defined a system with both unicast and multicast traffic and gave algorithms for practical calculations. Publication [3] studied how the signalling load due to multicast connections is distributed in the network. Aalto and Virtamo [10] created a convolution truncation algorithm for statistically indistinguishable channels, which was later generalised in Publication [4] for multiple groups of statistically indistinguishable channels, in Publication [5] to the case of two layer coding, and in Publication [6] to multi-layer coding. Publication [7] applied the simulation algorithm of Lassila and Virtamo [51] to the multicast setting, and Publication [8] generalised the algorithm to multi-layer case. Publication [9] studied multicast in a cellular network.

Our research, especially Publication [1] has triggered some activity on the area; the paper by Boussetta and Belyot [20] described above, and the papers of Rykov and Samouylov [65], Samouylov and Bobrikov [66] and Samouylov and Gaidamaka [67], which are mostly parallel to the papers [1] and [58]. Bosch i Pérez [18] studied three different methods for single link calculation, including the Polynomial distribution method.

2.8 Signalling

When dimensioning the network, signalling traffic needs also to be counted for. Signalling loads links with signalling messages, and control processors of the network elements. A review of signalling protocols in modern telecommunications and data networks was given in Kant and Ong [43]. Performance analysis of signalling networks was addressed in Bafutto *et al.* [17] and its references.

The processing load of a network node is composed of components that do not depend on traffic, such as adjacency protocols with neighbouring nodes, and components that depend on traffic, such as call setups and group management. The total processing load is the sum of these components.

Dynamic multicast connections need group management. The task of group management is to advertise groups to potential members and to control membership and various properties of the group. One possibility for this is to authorize the network nodes to establish a connection directly after receiving a request from a new member. This approach is available in ATM Forum User-Network Interface Signalling Specification (UNI) Version 4.0 [68], and IP multicast with Internet Group Management Protocol [37].

Another possibility is to have a special master node for group management. In this case, the master node initiates the new connections, and a network node willing to join the group may solicit initiation from the master node. Thus, all signalling traffic is relayed to the master node, which then signals to the relaying nodes the establishment of the new connection. A protocol with a master node is used by the MARS solution for IP over ATM [13, 15, 14]. The master node being a member of the multicast group is used in the Multicast Transport Protocol [16], and the master node being the root of the tree is used in ATM Forum UNI Version 4.0 [68].

In contrast to unicast traffic, in multicast traffic with a set of dynamic subtrees, signalling load varies depending on the observation point in the network. With unicast traffic, a traffic channel needs to be set up for each leg of the connection, while the number of legs affected with a user joining a multicast tree depends on the state of the tree.

This chapter presents the system model and the notation for the multicast loss system. Consider a network organized as a directed tree $G = (V, E)$, where V is the set of vertices, also called nodes, and E is the set of edges. The tree has a single root node with degree 1, and all the edges point away from it. The tree has $|E| = J$ edges and $|V| = J + 1$ nodes. The nodes are indexed from 1 to $J + 1$, i.e. $V = \{1, \ldots, J + 1\}$. Each edge $e \in E$ is an ordered pair $e = (i, j)$, where $i, j \in V$. Indexing of the nodes is chosen so that for each $e = (i, j) \in E$ it holds that $i > j$. Thus, the root node has index $J+1$. Each edge $e = (i, j)$ is uniquely defined by the node to which it points, j, and conversely for all nodes j except for the root node $J + 1$ there is a unique edge pointing to j. Edge $(i, j) \in E$ is also referred to as link j, and the set of links is denoted by $\mathcal{J} = \{1, \ldots, J\}$. Each link j has capacity $C_j \in \mathbb{N}$ resource units.

Let $\mathcal{N}_j$ denote the set of neighbouring links downstream of link j, i.e. $\mathcal{N}_j = \{i \in \mathcal{J} \mid (j, i) \in E\}$, and $\mathcal{M}_j$ denote the set of all links downstream of link j (including link j), i.e. the smallest subset of $\mathcal{J}$ such that $j \in \mathcal{M}_j$ and $\forall i \in \mathcal{M}_j$; $\mathcal{N}_i \subset \mathcal{M}_j$. See Figure 3.1 for an illustration of definitions.

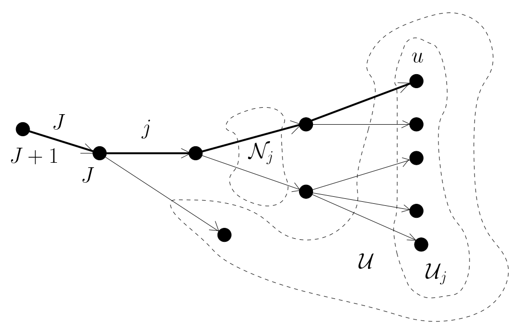

Figure 3.1: Network definitions. $\mathcal{R}_u$ is shown by the thick links.

The set $\mathcal{U} = \{j \in \mathcal{J} \mid \mathcal{N}_j = \emptyset\}$ denotes the set of the leaves of the tree. The leaf links and the user populations (nodes) connected to them are indexed with the same index $u \in \mathcal{U} = \{1, \ldots, U\}$. Note that this is possible without violating the indexing convention adopted. The user populations downstream of link j are denoted by $\mathcal{U}_j = \mathcal{U} \bigcap \mathcal{M}_j$. The size of the set $\mathcal{U}_j$ is denoted by U_j. The set of links on the route from link j to the root node is denoted by $\mathcal{R}_j$, defined as $\mathcal{R}_J = \{J\}$ and $\mathcal{R}_j = \mathcal{R}_i \bigcup \{j\}$ where $i \in \mathcal{J}$ and $(i, j) \in E$. The part of the path $\mathcal{R}_u$, $u \in \mathcal{U}$, that extends to the downstream neighbour of link j is denoted by $\mathcal{R}_u^j$ and is defined as $\mathcal{R}_u^j = \mathcal{R}_u \setminus \mathcal{R}_j$ if $j \in \mathcal{R}_u$ and $j, u \in \mathcal{J}$.

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. There are L layers, corresponding to different sets of substreams of hierarchically coded streams. Each layer $l \in \mathcal{L} = \{1, \ldots, L\}$ has a capacity requirement of

$d(l) \in \mathbb{N}$ capacity units, with $d(l) < d(l')$ for all $l < l'$, i.e. layer L contains all hierarchically coded sub-streams, layer 2 the two most important ones, and layer 1 contains only the most important sub-stream. In the multi-layer case, all channels do not necessary use all layers, and thus differing capacity requirements are possible. To allow different capacity requirements in the single layer case, let $\mathbf{c} = (c_1, \ldots, c_I)$ denote the capacity requirement vector for each channel in the single layer case, and c_i the capacity requirement for channel i.

3.1 State space

The states of the channels in a link define the state of that link. Each channel is in one of the states $\{0, 1, \ldots, L\}$, depending on whether the channel is *off*, or on layer $1, \ldots, L$. Denote the state of channel i on link j by the random variable $Y_{j,i} \in \{0, \ldots, L\}$, and the state of link j by the random vector $\mathbf{Y}_j = (Y_{j,i}; i \in \mathcal{I}) \in \mathcal{S} = \{0, \ldots, L\}^I$. The tuple (u, i, l) of the user population u (leaf node), channel i and layer l defines a multicast connection. The states $\mathbf{Y}_u$ of all the leaf links define the network state $\mathbf{X}$,

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

where $\Omega = \{0, \ldots, L\}^{U \times I}$ denotes the network state space. The network state determines the state of any link j as follows:

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

where $\max(\cdot)$ denotes the componentwise max-operation. The occupancy of any link j is determined by the link state as

$$D(\mathbf{Y}_j) = \sum_{i=1}^{I} d(Y_{j,i}),$$

where $d(0) = 0$, i.e. when channel is *off*, it does not need any link capacity. The occupancy generated by all other channels but I is denoted by $D'(\mathbf{Y}_j) = \sum_{i=1}^{I-1} d(Y_{j,i})$.

Now, in a finite capacity network, the capacity constraints of the links truncate the state space,

$$\tilde{\Omega} = \left\{ \mathbf{x} \in \Omega \,\middle|\, D(\mathbf{y}_j) \le C_j, \forall j \in \mathcal{J} \right\}.$$

3.2 Probability distributions

The user populations of the leaf links are assumed independent, and the leaf link distributions $\pi_u(\mathbf{y}) = \mathrm{P}\{\, \mathbf{Y}_u = \mathbf{y} \,\}$, $u \in \mathcal{U}$, are known, and represent stationary distributions of reversible Markov processes satisfying the detailed balance equations. The steady state probabilities $\pi(\mathbf{x})$ of the network states in a system with infinite link capacities can be calculated from

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

since the user populations are independent.

As already noted in Section 2.2, probabilities $\tilde{\pi}(\mathbf{x})$, $\mathbf{x} \in \tilde{\Omega}$, of states in a system with finite link capacities are obtained by truncation

$$\tilde{\pi}(\mathbf{x}) = \mathrm{P}\{\mathbf{X} = \mathbf{x} \,|\, \mathbf{X} \in \tilde{\Omega}\} = \frac{\pi(\mathbf{x})}{\mathrm{P}\{\, \mathbf{X} \in \tilde{\Omega} \,\}},$$

where $\mathrm{P}\{\, \mathbf{X} \in \tilde{\Omega} \,\} = \sum_{\mathbf{x} \in \tilde{\Omega}} \pi(\mathbf{x})$. This follows from the assumed detailed balance, and the co-ordinate convex nature of state space $\tilde{\Omega}$.

3.3 Time blocking probabilities

In a finite capacity network, blocking occurs whenever a user tries to establish a connection for channel i and layer r, and there is at least one link $j \in \mathcal{R}_u$ where the channel is on state $l < r$ and there is not enough spare capacity for setting the channel on the requested layer. Without loss of generality, and when not otherwise stated, the channels are ordered so that the blocking probability is calculated for channel with index I. Consider link j. A request for layer r is admitted if there is enough capacity already reserved for the layer in link j, or there is enough free capacity in the link, i.e.

$$\max\{d(r), d(y_{j,I})\} \leq C_j - D'(\mathbf{y}_j).$$

The expression "link j blocks" means that this condition does not hold for link j. The set $\mathcal{B}_{u,r}$ consists of the states where at least one link blocks for connection (u, I, r), when layer r of channel I is requested by user u, and is defined as

$$\mathcal{B}_{u,r} = \left\{ \mathbf{x} \in \tilde{\Omega} \,\middle|\, \exists j \in \mathcal{R}_u : d(r) > C_j - D'(\mathbf{y}_j) \right\}.$$

Then the time blocking probability for connection (u, I, r) is

$$B_{u,r} = \mathrm{P}\{\mathbf{X} \in \mathcal{B}_{u,r} \,|\, \mathbf{X} \in \tilde{\Omega}\} = \frac{\mathrm{P}\{\, \mathbf{X} \in \mathcal{B}_{u,r} \,\}}{\mathrm{P}\{\, \mathbf{X} \in \tilde{\Omega} \,\}}. \tag{3.1}$$

The model assumes that blocked calls are lost. This extends to the situation where there would be enough capacity for a lower layer connection; the user does not re-negotiate to get a lower layer connection, but starts a new idle period.

Calculation of time blocking probabilities for layers is possible, but time consuming: the number of states in the state space is of order $(L+1)^{UI}$. In addition, numerical problems may occur.

4 LINK-ORIENTED ANALYSIS

4.1 Multicast in a single link

Call blocking probabilities in a circuit switched network carrying multiple traffic classes can be calculated with exact algorithms, such as the recursion of [45] and [63], or with approximative methods, such as the normal type approximation [54]. These algorithms are applicable for unicast connections, such as telephone calls or ATM connections. They apply also for static multicast connections, where the structure of each multicast tree is fixed in advance. In a more dynamic environment, where the trees evolve with arriving and departing customers, these models are not adequate.

Nevertheless, they show a model for the first effort to calculate blocking probabilities. The approach is to create an algorithm to calculate call blocking probabilities in a single link and use the Reduced Load Approximation for the end-to-end blocking probabilities.

The Publications [1] and [2] use this approach. Publication [1] presents the single link blocking calculation and Publication [2] its application to end-to-end blocking probability calculation. These publications were followed by Boussetta and Belyot [20], who gave a product form solution for a case where there is both unicast and multicast traffic in the link.

This section describes the results of Publications [1] and [2] using the notation of the multicast loss system as defined in Chapter 3. The models presented in this section assume one layer ($L = 1$).

Network model

Even when considering a single link, several user populations may offer traffic for it. Thus, the network can be considered as having J links, where $C_j = C$ for link j and $C_{j'} = \infty$ for all $j' \neq j$. Then, links $j' \in \mathcal{J} \setminus \mathcal{M}_j$ do not contribute to blocking due to the infinite capacity, and thus user populations $u \in \mathcal{U} \setminus \mathcal{U}_j$ can be neglected. Similarly, links $j' \in \mathcal{M}_j \setminus \{j\}$ do not contribute to blocking events, and thus the topology of the part of the network downstream of link j is irrelevant. In other words, the remaining factors are the capacity of the link C_j, and the downstream user populations $\mathcal{U}_j$.

User model

The users downstream of the considered link subscribe to the multicast channels according a Poisson process with intensity λ. This is a model for an infinite user population, which is a reasonable assumption in networks with a large number of users, such as TV or radio multicasting in a network (for a link not too close to the leaves of the multicast tree). Further, each user chooses the channel independently of others and from the same preference distribution, α_i being the probability that channel i is chosen. As a result, the subscriptions to channel i arrive according to a Poisson process with intensity $\lambda_i = \alpha_i \lambda$. The users' holding times are generally distributed with mean $1/\mu_i$. Finally, let a_i denote the offered traffic intensity for channel i, $a_i = \lambda_i/\mu_i$.

Due to the Poisson assumption, the user populations $u \in \mathcal{U}_j$ can be combined simply by summing their call arrival intensities λ_u, so that $\lambda = \sum_{u \in \mathcal{U}_j} \lambda_u$.

Link occupancy in an infinite system

The single link model can be studied using the truncation principle, by first considering the link having an infinite capacity, and then truncating. The link state probabilities $\pi(\mathbf{y})$, where the state vector $\mathbf{y}$ is a vector of I values of 0 or 1, in the infinite system are as follows:

$$\pi(\mathbf{y}) = \prod_{i \in \mathcal{I}} p_i^{y_i} q_i^{1-y_i},$$

where p_i and q_i are found by inspecting the busy periods of an $M/G/\infty$ system, and calculated as follows:

$$p_i = 1 - e^{-a_i}, \qquad \text{and} \qquad q_i = e^{-a_i}.$$

All channels are independent from each other, and thus convolution, or, equivalently, the probability generating functions can be used to calculate the link occupancy distribution $Q(s) = \mathrm{P}\{\, D(\mathbf{Y}) = s \,\}$:

$$\sum_{s=0}^{\infty} Q(s) z^s = \prod_{i \in I} (q_i + p_i z^{c_i}).$$

Consider the on and off times of a single channel. Let $T_{i,\mathrm{on}}^{(\infty)}$ and $T_{i,\mathrm{off}}^{(\infty)}$ denote their means, respectively. These means can be calculated by using the $M/G/\infty$ system, too:

$$T_{i,\mathrm{on}}^{(\infty)} = \frac{e^{a_i} - 1}{\lambda_i}, \qquad \text{and} \qquad T_{i,\mathrm{off}}^{(\infty)} = \lambda_i^{-1}.$$

The mean on-time of the most popular channels as a function of the offered traffic intensity grows exponentially. This indicates that there is likely to be a set of channels that are almost constantly carried on the link.

Blocking in a link with finite capacity

It is important to make a distinction between various types of blocking. The *channel blocking probability* B_i^c of channel i is defined to be the probability that an attempt to turn channel i on fails due to insufficient capacity, whereas the *call blocking probability* b_i^c of channel i (seen by a user subscribing to channel i) refers to the probability that a user's attempt to subscribe to channel i fails. These are different, since the user's subscription is always accepted when the channel is already on. See Figure 4.1. Finally, the time blocking probability B_i^t, as noted in Chapter 2, of channel i is the probability that at least $C - c_i + 1$ capacity units of the link are occupied, and that channel i is off.

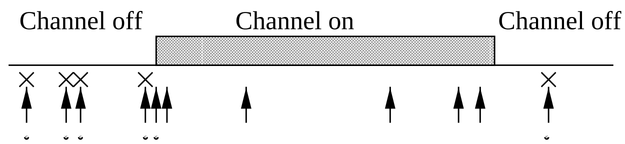

Figure 4.1: Call vs. channel blocking. Call attempts are represented by arrows, blocking events by crosses. The call attempts that are also attempts to turn a channel on are marked with dots. Call blocking in this trace is 5/11 and channel blocking 5/6.

The finite system can be described as a generalized Engset system. This refers to the well known $M/M/m/m/K$ system with a finite user population, see [47]. In a generalized Engset system the users are nonidentical, that is their mean holding and interarrival times as well as the requested resources can be different. Nevertheless, the stationary distribution of the system is insensitive with respect to the user idle and call holding time distributions. The call blocking probability in the system, however, is insensitive only to the call holding time distribution.

The multicast channels represent the users in the Engset system. When the channel is on, the 'user' is active, and when the channel is off, the 'user' is idle. Thus, the holding time of user i in the generalized Engset system is generally distributed with mean $T_{i,\text{on}}$, and the interarrival time is exponentially distributed with mean λ_i^{-1}. As a consequence, the channel blocking probability B_i^c equals the call blocking probability of user i in the corresponding generalized Engset system. Similarly, the time blocking probability B_i^t equals that of the generalized Engset system.

The time blocking probability of user i in the generalized Engset system can be calculated from the following formula:

$$B_i^t = \frac{\sum_{s=C-c_i+1}^{C} Q(s)}{\sum_{s=0}^{C} Q(s)},$$

where $Q(s)$ is the occupancy distribution in the infinite system. As noted in Section 2.3, the call blocking probability of user i equals the time blocking probability (of user i) in a system where user i is removed. Thus the channel blocking probability is as follows:

$$B_i^c = \frac{\sum_{s=C-c_i+1}^{C} Q'(s)}{\sum_{s=0}^{C} Q'(s)}, \tag{4.1}$$

where $Q'(s)$ is the occupancy distribution of an infinite link with channel i removed.

Note that the mean on-time of channel i is the same as in the infinite system, and that the mean off time is $T_{i,\text{off}} = 1/(\lambda_i(1 - B_i^c))$. Then, the call blocking probability of channel i is calculated as follows:

$$b_i^c = \frac{B_i^c}{(1 - B_i^c)(e^{a_i} - 1) + 1}. \tag{4.2}$$

To summarize, the call blocking probability b_i^c can be calculated from Equation (4.2) by using Equation (4.1). Note that the denominator in Equation (4.2) is always greater than 1. Thus, the call blocking probability b_i^c seen by a user subscribing to channel i is always smaller than the corresponding channel blocking probability B_i^c. This reflects the fact that a user subscribing to a channel while the channel is on never experiences any blocking. Furthermore, for the most popular channels, the blocking probability seen by a user drops practically to zero, since the exponential term in the denominator grows rapidly with a_i ($b_i^c \approx B_i^c e^{-a_i}$). For a channel with $a_i \ll 1$, the channel blocking probability and the call blocking probability seen by a user are approximately the same.

Since $b_i^c \leq B_i^c \leq B_i^t$, the time blocking probability in a system with all channels present is an upper bound for the call blocking probability. No call blocking probability seen by a user can be higher than this, but the call blocking probability for user approaches it for channels with channel preferences α_i near zero.

As a further comment, the above formulation of the call blocking probability is not specifically a multicast formulation. Examining the multicast loss system shows that the multicast blocking event for channel i has two conditions; first that the system is full, and second that channel i is not carried on the link. Noting this leads to the formulation of the call blocking probability given by Nyberg [56], where the call blocking probability b_i^c equals

$$b_i^c = b_i^t = \frac{\mathrm{P}\{\, \mathbf{X} \in \mathcal{B}_i \,\}}{\mathrm{P}\{\, \mathbf{X} \in \tilde{\Omega} \,\}} = \frac{q_i Q'(C)}{\sum_{s=0}^{C} Q(s)},$$

where b_i^t denotes the time blocking probability for channel i in the multicast loss system. Since the arriving calls follow a Poisson process, b_i^c and b_i^t equal.

Reduced Load Approximation

The Reduced Load Approximation can be applied to the multicast case by replacing the Erlang formula (or another link blocking formula) in the method with Equation (4.2) for multicast traffic.

To be more precise, let L_j^i denote the call blocking probability of channel i in link j. The method assumes that the links behave independently, or that the blocking events for each link are independent, so the probability that a call is not blocked on the route is calculated by multiplying the corresponding probabilities of individual links.

Let $a_{i,u}$ denote the traffic intensity offered by user population u for channel i. The blocking probabilities L_j^i are calculated from Equation (4.2),

$$L_j^i = b_i^c[\mathbf{r}_j, \mathbf{c}, C_j], \tag{4.3}$$

where the elements of the vector $\mathbf{r}_j$, $r_{j,i}$, represent the thinned traffic intensities for corresponding channels in the link j. They are calculated as follows:

$$r_{j,i} = \sum_{u \in \mathcal{U}_j} a_{i,u} \prod_{k \in \mathcal{R}_u - \{j\}} (1 - L_k^i). \tag{4.4}$$

Traffic intensities are calculated for each leaf node and each channel by thinning the offered intensities with the call blocking probabilities for all links along the route. The thinned traffic for a link and a channel is then given by the sum of these thinned traffic intensities divided by the link's own thinning factor.

Equations (4.3) and (4.4) form a fixed point equation of the form $\mathbf{L} = \mathbf{T}(\mathbf{L})$. The solution, a vector of L_j^i, may be found by repeated substitution. The equation might have several solutions [44]. (For the traditional RLA used for unicast calls, the solution is not always unique if capacity requirements of different traffic classes differ, see e.g. [64]) After the values for L_j^i have been found, the end-to-end blocking probability B_i^u for user u and channel i is calculated as:

$$B_i^u = 1 - \prod_{k \in R_u} (1 - L_k^i),$$

which follows directly from the assumption of independency.

Numerical results

In order to find out the accuracy of the Reduced Load Approximation in this multicast context, several examples were studied by both calculations using RLA and by simulations. The comparisons were carried out for two different network settings and several traffic intensities. The cases studied included cases where capacity requirements c_i of different channels are identical, $c_i = 1$ for all i, and one case where $c_i = 1$ if i is odd, and $c_i = 2$ if i is even.

As with using RLA with unicast traffic, the accuracy of the approximation declines as the traffic increases. This was to be expected since, with low traffic intensity, blocking is rare, and the traffic in each link is closer to Poisson traffic. When traffic intensity increases, blocking starts to shape traffic, which reduces real blocking. However, the model for single link blocking does not take this into account and gives higher blocking probabilities.

The error of RLA in a parking-lot type network is grows when the length of the route becomes longer. This can be traced back to the fact that the traffic processes in different links are not independent, since the traffic on the long route is dominating. That is, the consecutive links on the long route carry almost the same set of connections. Thus, the RLA's assumption of independence is not valid, introducing an error to the calculations.

The results show that the RLA yields results that are in the best cases almost equal and in the worst cases of the same order of magnitude as the true blocking probabilities. Since the blocking probabilities are usually "steep" functions of link capacity, an order of magnitude accurary in blocking probabilities does not introduce an excessive error in the link capacity. The estimated call blocking probabilities are also greater than the real ones, so that the approximation is conservative.

4.2 Multicast in a cellular network

Publication [9] considers blocking probabilities and dimensioning a single cell in a cellular network. To simplify the analysis, the interference of the neighbouring cells and other physical and technical constraints are assumed to be taken care of by specifying a capacity C (in connections) for the cell. The capacity corresponds to the maximum number of unicast connections in the unicast case, and the maximum number of ongoing multicast transmissions in the multicast case, i.e. the capacity requirement for all connections equals 1. Thus, the mobile cell is reduced to a single transmission link.

There are N homogenous users resident in the cell. Handovers are not treated separately; the handovers are assumed to behave as normal calls. According to the literature, the call holding times for handovers do not behave according to negative exponential distributions [42, 59, 24]. The handovers are left untreated; the main goal of the model is not in dimensioning but comparison of unicast and multicast, for which purpose a bit coarser approach is sufficient. For all finite I, however, the time blocking probabilities of the system are insensitive to the call holding time distribution. Each user behaves independently of the other users. The user may either be *active* or *inactive*. When "active", the user is engaged with an ongoing call. After a call is over, the user is inactive for a period of time. Assume that the times the user is inactive are exponentially distributed with parameter λ, and the call holding times are exponentially distributed with parameter μ. If a call is blocked, the user falls back and starts a new inactive period, i.e. blocked calls are lost.

Let p denote the probability that a user is active at any instant of time if there is no blocking in the system:

$$p = \frac{\lambda}{\lambda + \mu}.$$

Some channels are more popular than the others, and thus have often more simultaneous users. The *preference distribution* captures this behaviour. When starting a call, a user selects a channel using the preference distribution. The publication uses the Zipf distribution, as suggested by the studies of the Internet by Breslau *et al.* [21]. The point probabilities are of the form

$$a_i \propto i^{-\alpha}, \quad i = 1, 2, \ldots, \tag{4.5}$$

where α is a free parameter characterizing the distribution.

Multicast can provide content simultaneously to several users, thus multicast can be thought as more efficient than unicast. Multicast gain is then the ratio of the number of supported users by the cell by multicast to the corresponding unicast number. In order for multicast to give better efficiency compared with unicast there must be some channels that are clearly more popular than the others. In Zipf type distributions, Equation (4.5), the parameter α controls this phenomenon: the greater α, the more popular are the most popular channels (in comparison with the other channels). The work of Breslau *et al.* [21] indicates that α obtains larger values in a homogenous environment than when users are diverse. A similar result is found by Adamic and Huberman [11].

The multicast user, when in the inactive state, selects channel i with intensity $a_i\lambda$. After the call, the user returns to the inactive state. The

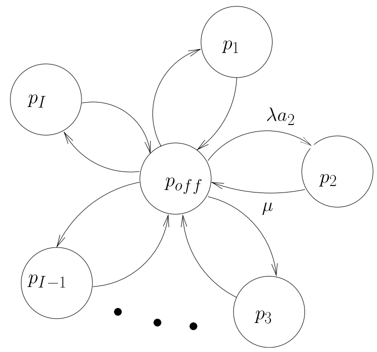

Figure 4.2: Multicast user model.

number of channels from which to select is denoted by I. The resulting user model is shown in Figure 4.2.

To be able to handle the case where the number of channels $I = \infty$, a Monte-Carlo simulation approach was chosen.

In the multicast setting, the preference distribution causes the call blocking probabilities for different channels to be different; if a channel is already carried on the link, there can be no blocking events for this channel. The call blocking probability is calculated from the channel-wise call blocking probabilities

$$B_{mc} = \sum_{i=1}^{I} a_i B_i^c,$$

where B_i^c is the channel-wise call blocking probability of channel i. Note that in the case where $I = \infty$, this sum has an infinite number of terms. For this case, if a_i are in a monotonically decreasing order, an upper bound is used;

$$\begin{aligned} B_{mc}^{+} &= \sum_{i=1}^{I_0} a_i B_i^c + B_\infty^c \sum_{i=I_0+1}^{I} a_i \\ &= F[N, p, C, \alpha, I, I_0], \end{aligned} \tag{4.6}$$

where I_0 can be selected so as to make the error insignificantly small. The blocking probabilities B_i^c and B_∞^c are explained in the next paragraph.

The channel-wise call blocking probability B_i^c for channel i can be simulated as the time blocking probability for channel i in a system where one user has been removed from the system, as discussed in Section 2.4. The time blocking probability is the probability of an event where there is no capacity left for channel i and channel i is not active. The call blocking probability B_∞^c can be thought of as the call blocking probability for a "channel that is never carried on a link", since its a_i is let to zero, and equals the time blocking probability of the system with one user removed.

The call blocking probability given by Equation (4.6) can be used for finding the required link capacity:

$$N_{mc} = \left\lfloor F^{-1}[B_{target}; p, C, \alpha, I, I_0] \right\rfloor .$$

The "inversion" can be done by iteratively simulating the call blocking probability and adjusting the number of users.

Simulation

The call blocking probabilities needed in Equation (4.6) can be simulated as follows.

1. The system state can be described by defining the states of the users $u \in \{1, \ldots, N'\}$, where $N' = N - 1$ (in most cases $N \ll I$). Thus, a vector of user states is generated by first drawing for each user whether they are active, and then drawing a channel number from the preference distribution for the active users.

2. The link occupancy X is calculated by counting the unique channel numbers for the users.

3. The resulting link occupancy X is examined:

   (a) If the link occupancy $X \leq C$, the sample is accepted. The number of accepted samples is increased, $n_{acc} := n_{acc} + 1$.

   (b) If $X = C$, the sample corresponds to a blocking state. The number of blocking state samples is increased, $n_B := n_B + 1$.

   (c) If the sample is a blocking state sample ($X = C$), the sample must be checked against the channels $i \leq I_0$. For all channels $i \leq I_0$ that are *not* active in the sample, the sample corresponds to a blocking state, and the number of channel-wise blocking state samples is increased, $n_B^i := n_B^i + 1$.

After a run of simulations, the channel-wise blocking probability B_i^c can be calculated as the ratio of the number of blocking state samples for channel i to the accepted samples, $B_i^c = n_B^i / n_{acc}$. The blocking probability B_∞^c is the ratio of the number of overall blocking state samples to the accepted samples, $B_\infty^c = n_B / n_{acc}$.

Numerical results

Publication [9] studies the ratio of the number of users that can be supported by multicast connections to the number of users that can be supported by unicast users, and calls this ratio multicasting gain. The simulations show that in order to get a significant multicasting gain, the parameter α of the preference distribution must be significantly higher than 1. Links with a higher capacity also favour multicast. This is due to the higher number of users supported by the cell, which increases the likelihood of simultaneous users on multicast channels.

As expected, when the number of available channels is smaller, the gain for multicast grows. However, the number of channels must be restricted to a rather low value to have a significant improvement.

If the user activity rises to a high level, part of the multicast gain is lost. This loss is higher with larger values of α. The natural explanation to this phenomenon is that for the values of p near 1, the number of users supported by the cell is smaller and thus the likelihood of simultaneous users of a multicast channel is again smaller.

The Internet studies predict that when users may freely choose content, the distribution parameter α will not have a very high value. Thus, intervention of the network operator is needed to secure a significant multicasting gain.

5 TREE-STRUCTURED NETWORKS

This chapter presents algorithms for exact evaluation of end-to-end blocking probabilities in multicast loss systems. First, a set of convolution-truncation algorithms are presented. These algorithms attack the computational complexity of the problem by grouping state probabilities into larger sets. Second, the inverse convolution algorithm for Monte-Carlo simulation is applied to the multicast loss system. Finally, some user population models are presented to allow calculating call blocking probabilities for users.

5.1 Convolution-truncation algorithms

The convolution-truncation algorithms presented in Publications [4], [5] and [6] were all derived for attacking specific situations where structure of the multicast loss system and possible common statistical properties of the multicast channels could be exploited. All of these algorithms are successors of the algorithm by Nyberg *et al.* [58]. The MAX-convolution algorithm is a straightforward generalisation of the algorithm of Nyberg *et al.* to the case where multicast streams are layered. Publication [4] studies the case where some multicast channels are statistically indistinguishable from each other. This means that

(i) these channels are chosen with the same probabilities

(ii) the mean holding time for these channels is the same, and

(iii) the capacity needed to carry any of these channels on any link is the same.

In this case, a reduction in the computational complexity is available. Publications [5] and [6] present an algorithm for calculating blocking probabilities in the case where all channels are statistically indistinguishable, but the channels use layered coding. This section presents a more general framework in which all these algorithms fit, i.e. the intention is to show the common structure of all these algorithms.

Binary trees

In order to simplify calculations, the multicast loss system is first transformed into a tree where all nodes have at most two child links (for all $j \in \mathcal{J}\ ;\ |\mathcal{N}_j| \leq 2$). This can be done by introducing supplementary links.

Supplementing means the process of adding augmented links with infinite capacity to yield a binary tree. There are many possible ways to supplement trees, see Figure 5.1 for an example. The states of the original network are the same as the states in the corresponding links in the supplemented network. Because of this property, every algorithm that works properly in the supplemented network, works in any multicast loss system.

Given an original tree $\mathcal{J}$ and any supplemented tree $\mathcal{J}'$, the state of any link j in the original network can be calculated recursively from the

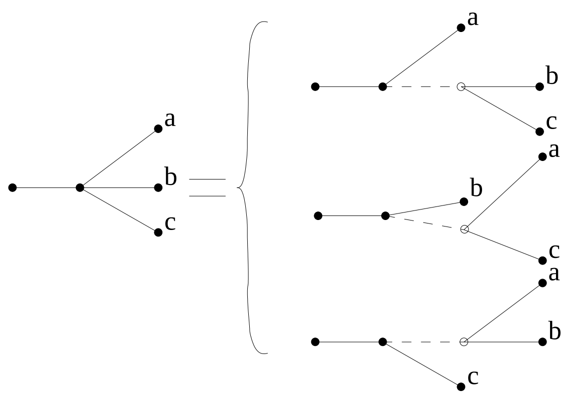

Figure 5.1: Original network and possible supplemented networks. In this case, there are three possible supplemented networks, each having exactly the same states in the links from the original network (solid lines).

leaf links $\mathcal{U}_j$ using the supplemented tree as

$$\mathbf{Y}_j = \begin{cases} \mathbf{Y}_u & \text{if } j = u \in \mathcal{U}, \\ \max(\mathbf{Y}_{j'}, \mathbf{Y}_{j''}) & \text{otherwise,} \end{cases}$$

where $\mathcal{N}'_j = \{j', j''\}$, and $\mathcal{N}'_j$ is the set of downstream neighbour links of link j in the supplemented tree.

This result follows directly from the associativity and commutativity of the $\max(\cdot)$ operation, applied recursively to the whole tree. In the rest of this section, all multicast loss systems are supplemented trees.

Convolution and truncation operations

In order to make calculations more efficient, the algorithms presented in this section use groups of link states instead of network states. For this purpose, a concept called *partial state descriptor* is defined. The partial state descriptor space Φ is a partition of the link state space $\mathcal{S}$, and the partial state descriptor $\phi \in \Phi$ is thus a subset of the state space $\mathcal{S}$. The partitioning is done so that for each link state $\mathbf{y} \in \phi$, the link occupancy $D(\mathbf{y})$ is equal, and is denoted by $D(\phi)$. Note that this condition does not prohibit several partial state descriptors ϕ producing the same link occupancy. Let $\pi_j(\phi)$ denote the probability $\mathrm{P}\{\, \mathbf{Y}_j \in \phi \,\}$, i.e. the probability that link j is in one of the link states in ϕ.

Some of the partial state descriptors to be used utilise the fact that in some cases some channels are statistically indistinguishable, and can be grouped. Let K denote the number of groups of channels. Let $\mathbf{I} = (I_1, \ldots, I_K)$ denote a vector whose elements correspond to the number of channels in each group. Naturally, they sum up to I, $\mathbf{I} \cdot \mathbf{1} = I$. The groups are ordered so that the considered channel number I belongs to group number K. Let $\mathbf{I}' = \mathbf{I} - \mathbf{e}_K$, i.e. the number of channels in each group, where channel I is removed.

In order to exploit the properties of the different scenarios, the presented algorithms use the following partial state descriptor spaces:

1. Full state description:

$$\Phi = \{\{\mathbf{y}\} |\, \mathbf{y} \in \mathcal{S}\}.$$

2. The number of channels n that are on (when all channels are statistically indistinguishable), in a single layer model:

$$\Phi = \{\{\mathbf{y} |\, \mathbf{y} \cdot \mathbf{1} = n\} |\, n \in \mathbb{N},\, n \leq I\},$$

where each value of n corresponds to a unique $\phi \in \Phi$.

3. The number of channels $\mathbf{n}$ that are on (when channels belong to K groups of statistically indistinguishable channels), in a single layer model:

$$\Phi = \left\{ \left\{ \mathbf{y} \middle| \sum_{i \in \mathcal{I}} y_i 1_{i \in \mathcal{I}_k} = n_k,\, \forall k \right\} \middle| \mathbf{n} \in \mathbb{N}^K,\, \mathbf{n} \cdot \mathbf{1} \leq I,\, n_k \leq I_k \forall k \right\},$$

where $\mathcal{I}_k$ denotes the set of channels in group number k of statistically indistinguishable channels, and 1_{cond} the indicator function, which returns 1 if cond, and 0 otherwise. Each $\mathbf{n}$ defines a unique $\phi \in \Phi$.

4. The number of channels $\mathbf{k}$ that are on each layer $l \in \mathcal{L}$ (when all channels are statistically indistinguishable)

$$\Phi = \left\{ \left\{ \mathbf{y} \middle| \sum_{i \in \mathcal{I}} 1_{y_i = l} = k_l,\, \forall l \right\} \middle| \mathbf{k} \in \mathbb{N}^L \,,\, \mathbf{k} \cdot \mathbf{1} \leq I \right\},$$

where each $\mathbf{k}$ defines a unique $\phi \in \Phi$.

Each particular link j has a finite capacity of C_j, which constrains the allowed link state space. The corresponding partial state descriptor space $\tilde{\Phi}_j$ is defined as

$$\tilde{\Phi}_j = \{\phi | \phi \in \Phi \,,\, D(\phi) \leq C_j\}.$$

Calculating the probability mass of set $\tilde{\Omega} \backslash \mathcal{B}$ requires a special treatment for channel I. For this purpose, most algorithms need a second partial state descriptor definition. Let φ denote a partial state descriptor defined with channel I removed. Then, the state of channel I, y_I, is explicitly introduced to the partial state descriptor. Let $F(\varphi, l)$ denote the partial state descriptor

$$F(\varphi, l) = \{\mathbf{y} |\, \mathbf{y}' \in \varphi,\, y_I = l\} \in \Phi,$$

where $\mathbf{y}'$ is the link state without channel I. Then,

$$\Phi = \{F(\varphi, l) | l = 0, 1, \ldots, L \,,\, \varphi \in \Phi'\} \,,$$

where Φ' denotes the partial state descriptor space without channel I. Let $D'(\varphi)$ denote the link occupancy generated by the states in partial state descriptor φ. Let $\tilde{\Phi}_{j,r}$ denote the set of partial state descriptors that do not violate the capacity restriction in link j, and where there is capacity free for a request of capacity r (or channel I is already on a layer higher than r):

$$\tilde{\Phi}_{j,r} = \{F(\varphi, l) |\, D'(\varphi) + \max(d(l), d(r)) \leq C_j\}.$$

In a tree-structured loss system, the state of the system is defined by the states of the leaf links, and the capacity constraints are defined by the links. The algorithms exploit this by recursively using the truncation and convolution operations. The convolution-truncation algorithms use a successive series of operations called convolutions to group link state information when moving up the tree, and truncation operations to force link capacity constraints (i.e. leave out groups of states not in the allowable state space). This is possible by (according to the truncation principle) calculating the state probabilities from a system with no capacity restrictions and then normalising them. All leaves of the tree are independent prior to truncation, and thus a product form solution for the probabilities exist. Truncation can be made in several parts, and it is possible to make a product form solution of the truncated probabilities, as far as they are mutually independent.

First, let the notation $[\phi', \phi''] = \phi$ mean that ϕ' and ϕ'' get all values that satisfy for all $\mathbf{y}' \in \phi'$ and $\mathbf{y}'' \in \phi''$ the common link state $\mathbf{y} = \max(\mathbf{y}', \mathbf{y}'') \in \phi$. Then, two different convolution operations are defined, one for the denominator and one for the numerator of Equation (3.1):

Definition 5.1.1 *The convolution operation* $\otimes$ *for any two functions* $f, g : \Phi \mapsto \mathbb{R}$ *is defined as*

$$[f \otimes g](\phi) = \sum_{[\phi', \phi'']=\phi} f(\phi') g(\phi'') \varsigma(\phi, \phi', \phi'', \mathbf{I}),$$

where $\phi, \phi', \phi'' \in \Phi$*, and* $\varsigma(\cdot)$ *is an algorithm specific real-valued function.*

Definition 5.1.2 *The convolution operation* $\odot$ *for any two functions* $f, g : \Phi \mapsto \mathbb{R}$ *is defined as*

$$[f \odot g](\varphi, l) = \sum_{[\varphi', \varphi'']=\varphi} \sum_{\substack{v', v'' \\ \max(v', v'') = l}} f(\varphi', v') E(g, \varphi'', v'') \times \varsigma(F(\varphi, l), F(\varphi', v'), F(\varphi'', v''), \mathbf{I}'),$$

where a mapping $E(\cdot)$*, is defined as*

$$E(f, \varphi, l) = \begin{cases} \left(1 - \frac{\boldsymbol{\nu}(\varphi) \cdot \mathbf{1}}{I_K}\right) f(F(\varphi, 0)) & \text{if } l = 0. \\ \left(\frac{\nu_l(\varphi)+1}{I_K}\right) f(F(\varphi, l)) & \text{if } l > 0. \end{cases}$$

where $\boldsymbol{\nu}(\varphi)$ *is a vector of the number of channels that are on each layer and belong to group* K *(excluding channel* I*).*

The mapping $E(\cdot)$ can be thought of as introducing a combinatorial coefficient giving the ratio of channels belonging to group K that are on the same layer as channel I, including channel I, to the number of channels in that group, I_K.

To force the calculation in $\tilde{\Omega}$, a second operation, called truncation, is used. The operation is defined separately for the denominator and for the numerator:

Definition 5.1.3 *The truncation operation for functions* $f : \Phi \mapsto \mathbb{R}$ *for the denominator is defined as*

$$T_j^d f(\phi) = f(\phi) 1_{D(\phi) \leq C_j}.$$

Definition 5.1.4 *The truncation operation for functions* $f : \Phi \mapsto \mathbb{R}$ *for the numerator is defined as*

$$T_{j,r}^n f(\varphi, l) = f(\varphi, l) 1_{D'(\varphi) + \max(d(r), d(l)) \leq C_j}.$$

Now, the algorithm can be summarised as follows:

- The terms in the state sum $\mathrm{P}\{\, \mathbf{X} \in \tilde{\Omega} \,\}$ (Equation (3.1))

$$\mathrm{P}\{\, \mathbf{X} \in \tilde{\Omega} \,\} = \sum_{\phi \in \tilde{\Phi}_J} Q_J^d(\phi),$$

  are calculated recursively as

$$Q_j^d(\phi) = \begin{cases} T_j^d \pi_j(\phi) & \text{if } \mathcal{N}_j = \emptyset, \\ T_j^d Q_s^d(\phi) & \text{if } \mathcal{N}_j = \{s\}, \text{ and} \\ T_j^d \left[Q_s^d \otimes Q_t^d \right](\phi) & \text{if } \mathcal{N}_j = \{s, t\}. \end{cases}$$

- The terms in the state sum $\mathrm{P}\{\, \mathbf{X} \in \tilde{\Omega} \setminus \mathcal{B}_{u,r} \,\}$ (Equation (3.1))

$$\mathrm{P}\{\, \mathbf{X} \in \tilde{\Omega} \setminus \mathcal{B}_{u,r} \,\} = \sum_{(\varphi, l) \in \tilde{\Phi}'_{J,r}} Q_{J,r}^n(\varphi, l),$$

  are calculated recursively as

$$Q_{j,r}^n(\varphi, l) = \begin{cases} T_{j,r}^n \pi_j(F(\varphi, l)) & \text{if } \mathcal{N}_j = \emptyset, \\ T_{j,r}^n Q_{s,r}^n(\varphi, l) & \text{if } \mathcal{N}_j = \{s\}, \text{ and} \\ T_{j,r}^n \left[Q_{s,r}^n \odot Q_t^d \right](\varphi, l) & \text{if } \mathcal{N}_j = \{s, t\} \text{ and } s \in \mathcal{R}_u. \end{cases}$$

The complexity of the algorithm is seen to be linear with respect to the number of leaf links U, for all networks where for each internal link $j \in \mathcal{J} \setminus \mathcal{U}$ there are at least two child nodes, $|\mathcal{N}_j| \geq 2$. Trees with links having only one child link need some extra truncation operations, but generally, convolution operations are more laborious than truncations. This is a significant improvement compared to the exponential complexity of the original problem.

Background traffic

In common networking scenarios, the multicast trees are not in isolation, but the multicast routing tree can be seen as a virtual network over the real one. There is other traffic in the network, including other multicast trees (with different routing trees) and unicast transmissions. For an approximative treatment of other than the multicast tree traffic, a background traffic approach was developed in [58]. It assumes that the background traffic is independent between links, and thus the truncations in the truncation convolution algorithm can still be executed as before.

The independent background traffic approach is easily applied to the class of truncation-convolution algorithms. First, calculate the link occupancy distribution $q_j(z)$ for occupancy z in link j in the case there is *only* background traffic. Then, redefine the truncation operations as follows:

Definition 5.1.5 *Define truncation operator* $\widehat{T}_j^d$ *for any function* $f : \Phi \mapsto \mathbb{R}$ *as*

$$\widehat{T}_j^d f(\phi) = \left(\sum_{z=0}^{C_j - D(\phi)} q_j(z) \right) f(\phi).$$

Definition 5.1.6 *Define truncation operator* $\widehat{T}_{j,r}^n$ *for any function* $f : \Phi \mapsto \mathbb{R}$ *as*

$$\widehat{T}_{j,r}^n f(\varphi, l) = \left(\sum_{z=0}^{C_j - D'(\varphi) - \max\{d(r), d(l))\}} q_j(z) \right) f(\varphi, l).$$

In both of these definitions, an empty sum is equal to 0.

Algorithm steps

To calculate the time blocking probability $B_{u,r}$, state sums $\mathrm{P}\{\,\mathbf{X} \in \tilde{\Omega} \setminus \mathcal{B}_{u,r}\,\}$ and $\mathrm{P}\{\,\mathbf{X} \in \tilde{\Omega}\,\}$ are needed. These are calculated using a convolution-truncation algorithm, summarised as follows:

1. Set $j := 1$.
2. Calculate $Q_j^d(\phi)$ for all $\phi \in \tilde{\Phi}_j$.
3. If $j \in \mathcal{R}_u$, calculate $Q_{j,r}^n(\phi)$ for all $\phi \in \tilde{\Phi}_{j,r}$.
4. Set $j := j + 1$. If $j \leq J$, jump to step 2.
5. Calculate the sum of $Q_{J,r}^n(\phi)$ for all $\phi \in \tilde{\Phi}_{J,r}$ and divide it with the sum of $Q_J^d(\phi)$ for all $\phi \in \tilde{\Phi}_J$ to get $1 - B_{u,r}$.

Note here that the link numbering scheme ($\forall j' \in \mathcal{N}_j; j' < j$) guarantees that link probabilities for all links $j' \in \mathcal{N}_j$ have already been calculated when calculation of the link probability of link j starts.

Specific algorithms

This subsection shortly describes the algorithms presented in Publications [4], [5] and [6]. The algorithms are summarised in Table 5.1.

The MAX-convolution algorithm in Publication [6] is a straightforward generalisation of the convolution-truncation algorithm by Nyberg *et al.* [58]. The partial state descriptor used is the full state description. Since all channels are treated individually, no grouping of channels is needed.

Table 5.1: Convolution-truncation algorithms

Algorithm	Q_j^d	$\varsigma(I)$
MAX-conv.	$\mathrm{P}\{\mathbf{Y}_j = \mathbf{y}\ ;\ D(\mathbf{Y}_{j'}) \leq C_{j'}, j' \in \mathcal{M}_j\}$	—
1-layer single group	$\mathrm{P}\{N_j = n\ ;\ N_{j'} \leq C_{j'}, \forall j' \in \mathcal{M}_j\}$	$\frac{\binom{l}{l+m-n}\binom{I-l}{n-l}}{\binom{I}{m}}$
1-layer multi-group	$\mathrm{P}\{\mathbf{N}_j = \mathbf{n}\ ;\ \mathbf{N}_{j'} \cdot \mathbf{c} \leq C_{j'}, \forall j' \in \mathcal{M}_j\}$	$\prod_{k \in \mathcal{K}} \frac{\binom{l_k}{l_k+m_k-n_k}\binom{I_k-l_k}{n_k-l_k}}{\binom{I_k}{m_k}}$
L-layer single group	$\mathrm{P}\{\mathbf{K}_j = \mathbf{k}\ ;\ D(\mathbf{K}_{j'}) \leq C_{j'}, j' \in \mathcal{M}_j\}$	$\sum_{\mathbf{x}} \frac{\binom{I-\sum_{a\geq 1} l_a}{m_1-\sum_{a\geq 1} x_{a,1}\ \cdots\ m_L-\sum_{a\geq 1} x_{a,L}}}{\binom{I}{m_1\ \cdots\ m_L}} \times \prod_{a=1}^{L} \binom{l_a}{x_{a,1}\ \cdots\ x_{a,L}}$

Algorithm	$Q_{j,r}^n$	Complexity
MAX-conv.	$\mathrm{P}\{\mathbf{Y}_j = \mathbf{y}\ ;\ D(\mathbf{Y}_{j'}) \leq C_{j'}, j' \in \mathcal{M}_j\ ;\ d(r) \leq C_{j'} - D'(\mathbf{Y}_{j'}), j' \in \mathcal{M}_j \cap \mathcal{R}_u\}$	$O((U-1)(L+1)^{2I})$
1-layer single group	$\mathrm{P}\{N'_j = n\ ;\ N'_{j'} \leq C_{j'} - 1, \forall j' \in \mathcal{M}_j \cap \mathcal{R}_u\ ;\ N_{j'} \leq C_{j'}, \forall j' \in \mathcal{M}_j \setminus \mathcal{R}_u\}$	$O((U-1)C^3)$
1-layer multi-group	$\mathrm{P}\{\mathbf{N}'_j = \mathbf{n}\ ;\ \mathbf{N}'_{j'} \cdot \mathbf{c} \leq C_{j'} - c_K, \forall j' \in \mathcal{M}_j \cap \mathcal{R}_u\ ;\ \mathbf{N}_{j'} \cdot \mathbf{c} \leq C_{j'}, \forall j' \in \mathcal{M}_j \setminus \mathcal{R}_u\}$	$O((U-1)\prod_k(\min(I_k, C)+1)^3)$
L-layer single group	$\mathrm{P}\{\mathbf{K}'_j = \mathbf{k}, L_j = l\ ;\ D(\mathbf{K}'_{j'}) + d(r) \leq C_{j'}, j' \in \mathcal{M}_j \cap \mathcal{R}_u\ ;\ D(\mathbf{K}_{j'}) \leq C_{j'}, j' \in \mathcal{M}_j\}$	$O((U-1)(I+1)^{L(L+2)})$

The combinatorial convolution algorithm, developed by Aalto and Virtamo in [10] and reproduced in Publication [4], assumes that the channels can be considered statistically indistinguishable, or they behave in the same manner. In this algorithm, it is no longer necessary to keep track of all the channels individually, but instead of the number of channels. There is no need for specific capacity demands for channels, since they are equal and can be normalised to 1. The algorithm (as presented in Publication [4]), uses $C' = C - 1$ for calculating the numerator. The function $\varsigma(\cdot)$ is used to calculate the number of possible l and m channels taken from I that result in n channels in the common link.

For a multi-group combinatorial convolution, presented also in Publication [4], the multicast channels are divided into groups of statistically indistinguishable channels. When the number of groups grows, the complexity of the algorithm approaches that of the MAX-convolution algorithm. All channels within a group are similar, and thus the partial state descriptors used correspond to vectors of the number of channels *on* in each group. The algorithm (as presented in Publication [4]) also uses $C' = C - c_K$ for calculating the numerator, as the single group algorithm, and uses a product of the same $\varsigma(\cdot)$ function as the single group algorithm.

Publications [5] and [6] present a combinatorial algorithm for a single group of statistically indistinguishable channels using layered coding.

The algorithm of Publication [5] allows calculations for two-layer coding, and Publication [6] for multiple layers. In the publications, the partial state descriptors used correspond to vectors $\mathbf{k}$ of the number of channels on each layer. The function $\varsigma(\cdot)$ is more complicated than for the single-layer case, since there is a significantly larger number of intersections $\mathbf{X}$ for the different sets of channels (see Publication [6] for more detail). It is straightforward to generalise the multi-layer algorithm to the multi-group case, but the computational effort needed by the algorithm might hinder the practical use of it.

5.2 Simulation using Inverse Convolution

Another possible approach for solving blocking probabilities in multicast loss systems is to use simulations. As the form of the stationary distribution is known, the Monte Carlo (MC) method can be used. In order to make the simulation more efficient, it is possible to use importance sampling (IS), i.e. use an alternative sampling distribution making the interesting samples more likely than under the original distribution. The error introduced by the distribution change is then corrected by weighting the samples with the so called likelihood ratio.

The use of IS in MC estimation of blocking probabilities has been previously studied in [50, 64, 51]. The loss system studied in these works is the multiservice loss system. The multicast loss system studied in this dissertation, however, has sufficiently many common features with the multiservice loss system to allow application of the so called inverse convolution method developed by Lassila and Virtamo [51]. Publications [7] and [8] show the application of the inverse convolution algorithm to the multicast loss system, in the single layer case and in the multi-layer case. This section presents the results of Publication [8], noting that the single layer case is a special case of the multi-layer case.

The approach decomposes the problem into independent subproblems. The decomposition corresponds to dividing the blocking probability into components each of which essentially gives the blocking probability contribution from a single link. Then an efficient IS distribution is used to estimate the blocking probability contribution from each link. The distribution is a conditional distribution and allows generating samples directly from the set of blocking states of a given link, assuming that only the considered link has a finite capacity (cf. the single link case in Section 4.1). The inverse convolution method is then used to generate these samples. The method requires that channels are independent from each other in the infinite system (i.e. system where capacity constraints have been removed). This condition restricts the usable user population models. Some user population models are treated in Section 5.3.

The main problem in the simulation is to quickly get a good estimate for $\mathrm{P}\{\, \mathbf{X} \in \mathcal{B}_{u,r} \,\}$, i.e. the numerator in Equation (3.1), especially in the case when the blocking probability $B_{u,r}$ is small. Note that $B_{u,r}$ also depends on $\mathrm{P}\{\, \mathbf{X} \in \tilde{\Omega} \,\}$ given by the denominator of Equation (3.1). This probability is usually close to unity and is easy to estimate using the standard Monte Carlo method. Therefore, the publications concentrate on efficient methods for estimating $\mathrm{P}\{\,\mathbf{X} \in \mathcal{B}_{u,r}\,\}$.

Decomposition

In order to divide the task of estimating $P(\mathcal{B}_{u,r})$ to simpler sub-problems, $\mathcal{B}_{u,r}$ is partitioned into sets $\mathcal{E}^j_{u,r}$, which are defined as the sets of states in $\mathcal{B}_{u,r}$ where link j blocks but none of the links closer to user u block,

$$\mathcal{E}^j_{u,r} = \mathcal{B}_{u,r} \cap \Big\{ \mathbf{x} \in \tilde{\Omega} \,\Big|\, d(r) > C_j - D'(\mathbf{y}_j) \wedge \\ d(r) \le C_{j'} - D'(\mathbf{y}_{j'}), \forall j' \in \mathcal{R}^j_u \Big\}.$$

The $\mathcal{E}^j_{u,r}$ form a partitioning of $\mathcal{B}_{u,r}$, i.e.

$$\mathcal{B}_{u,r} = \bigcup_{j \in \mathcal{R}_u} \mathcal{E}^j_{u,r},$$

and $\mathcal{E}^j_{u,r} \cap \mathcal{E}^{j'}_{u,r} = \emptyset$, when $j \ne j'$. Thus,

$$\mathrm{P}\{\,\mathbf{X} \in \mathcal{B}_{u,r}\,\} = \sum_{j \in \mathcal{R}_u} \mathrm{P}\{\,\mathbf{X} \in \mathcal{E}^j_{u,r}\,\}. \tag{5.1}$$

The probability $\mathrm{P}\{\,\mathbf{X} \in \mathcal{E}^j_{u,r}\,\}$ can be thought of as the blocking probability contribution due to link j. It should be noted, however, that blocking in the states where several links block can be arbitrarily attributed to any of the blocking links. The definition used here is based on a convention which attributes it to the blocking link closest to the user.

Conditioning of $\mathrm{P}\{\,\mathbf{X} \in \mathcal{E}^j_{u,r}\,\}$

Equation (5.1) decomposes the estimation of $\mathrm{P}\{\,\mathbf{X} \in \mathcal{B}_{u,r}\,\}$ into independent sub-problems of estimating the $\mathrm{P}\{\,\mathbf{X} \in \mathcal{E}^j_{u,r}\,\}$. For these estimation tasks, the superset $\mathcal{D}^j_{u,r} \supset \mathcal{E}^j_{u,r}$ is introduced,

$$\mathcal{D}^j_{u,r} = \Big\{ \mathbf{x} \in \Omega \,\Big|\, d(r) > C_j - D'(\mathbf{y}_j) \ge d(y_{j,I}) \Big\}.$$

This set corresponds to blocking states in a system where link j has a finite capacity C_j but all other links have infinite capacity. Since all links have finite capacity in real systems, and several links could block simultaneously, sets $\mathcal{D}^j_{u,r}$ are not disjoint unlike their subsets $\mathcal{E}^j_{u,r}$.

The next step is to use conditional probabilities to estimate $\mathrm{P}\{\,\mathbf{X} \in \mathcal{E}^j_{u,r}\,\}$, as follows:

$$\mathrm{P}\{\,\mathbf{X} \in \mathcal{E}^j_{u,r}\,\} = \mathrm{P}\{\mathbf{X} \in \mathcal{E}^j_{u,r} \,|\, \mathbf{X} \in \mathcal{D}^j_{u,r}\}\mathrm{P}\{\,\mathbf{X} \in \mathcal{D}^j_{u,r}\,\}.$$

This equation can be thought of as the importance sampling equation, where instead of the original distribution (of $\mathcal{E}^j_{u,r}$) a conditional distribution is used, and the result is corrected with the likelihood ratio $\mathrm{P}\{\,\mathbf{X} \in \mathcal{D}^j_{u,r}\,\}$. This is useful from the simulation point of view since it is easy to compute

$\mathrm{P}\{\, \mathbf{X} \in \mathcal{D}^j_{u,r} \,\}$ and to generate samples from the original distribution under the condition $\mathbf{X} \in \mathcal{D}^j_{u,r}$, as explained later. Monte Carlo simulation is then used to estimate the conditional probability $\mathrm{P}\{\mathbf{X} \in \mathcal{E}^j_{u,r} \,|\, \mathbf{X} \in \mathcal{D}^j_{u,r}\}$ instead of $\mathrm{P}\{\, \mathbf{X} \in \mathcal{E}^j_{u,r} \,\}$, which is usually much more effective.

Let $\widehat{\eta}^j_{u,r}$ denote the estimator for $\eta^j_{u,r} = \mathrm{P}\{\, \mathbf{X} \in \mathcal{E}^j_{u,r} \,\}$,

$$\widehat{\eta}^j_{u,r} = \frac{v_j}{N_j} \sum_{n=1}^{N_j} 1_{\mathbf{X}^*_n \in \mathcal{E}^j_{u,r}}, \tag{5.2}$$

where $v_j = \mathrm{P}\{\, \mathbf{X} \in \mathcal{D}^j_{u,r} \,\}$ and $\mathbf{X}^*_n$ denotes samples drawn from the conditional distribution $\mathrm{P}\{\mathbf{X} = \mathbf{x} \,|\, \mathbf{X} \in \mathcal{D}^j_{u,r}\}$. Then, the estimator for $P(\mathcal{B}^j_{u,r})$ is simply

$$\widehat{P}(\mathcal{B}^j_{u,r}) = \sum_{j \in \mathcal{R}_u} \widehat{\eta}^j_{u,r}.$$

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 chosen to minimise the variance of the estimate. This is done by assigning the number of samples to different $\widehat{\eta}^j_{u,r}$ according to their estimated variance during the simulation, for details see Publication [7].

Algorithm

Consider only the estimation of $\eta^j_{u,r}$ for fixed $j \in \mathcal{R}_u$ and traffic class (u, I, r). The method is based on generating points from the conditional distribution $\mathrm{P}\{\mathbf{X} = \mathbf{x} \,|\, \mathbf{X} \in \mathcal{D}^j_{u,r}\}$ by reversing the steps used to calculate the occupancy distribution of the considered link. Note that the condition $\mathbf{X} \in \mathcal{D}^j_{u,r}$ is a condition expressed in terms of the occupancy, $D'(\mathbf{y}_j)$, 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. This procedure is explained later. Then, given the state $\mathbf{Y}_j$, the state of the network, i.e. states of the leaf links, is generated. The mapping $\mathbf{x} \mapsto \mathbf{y}_j$ 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 (See Fig. 5.2.):

1. Generate the states for leaf links u by
   (a) Generate a sample state $\mathbf{Y}_j$ under the condition $d(r) > C_j - D'(\mathbf{y}_j) \geq d(y_{j,I})$ for link j.
   (b) Generate the leaf link states $\mathbf{Y}_u$, $u \in \mathcal{U}_j$, with the condition that link j state $\mathbf{Y}_j = \max_{u \in \mathcal{U}_j}(\mathbf{Y}_u)$ 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 $\mathbf{X}^*_n \in \mathcal{D}^j_{u,r}$ consists of the set of all sample states of leaf links generated with step 1.
3. To collect the statistics for estimator $\widehat{\eta}^j_{u,r}$, check if $\mathbf{X}^*_n \in \mathcal{E}^j_{u,r}$.

The above steps are repeated for generating N_j samples.

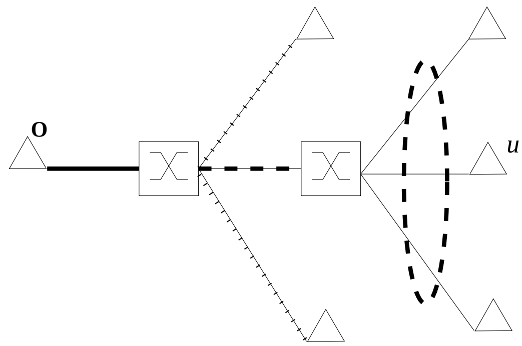

Figure 5.2: Example of sample generation. A sample in the set $\mathcal{D}$ is generated for the link denoted by the thick dashed line. States of the links marked by the dashed ellipse are generated by inverse convolution from the state. States for links denoted by ticks are generated by a simple draw. State of the link denoted by the thick line is calculated directly from the states of the other links.

Generating the link state

First, the link occupancy $D(\mathbf{Y}_j)$ is easily calculated recursively as follows. Let $S_{j,i}$ denote link occupancy due to the first i channels,

$$S_{j,i} = \sum_{i' \leq i} d(Y_{j,i'}).$$

Then $D(\mathbf{Y}_j) = S_{j,I}$ and $D'(\mathbf{Y}_j) = S_{j,I-1}$. The $Y_{j,i}$ are mutually independent, and $S_{j,i} = S_{j,i-1} + d(Y_{j,i})$, where $S_{j,i-1}$ and $Y_{j,i}$ are independent.

Channel I must be dealt with differently from the other channels, since the system can be in a blocking state only if $C_j - S_{j,I-1} < d(r)$, but the channel I can be in any state $l < r$. Knowing this, the set $\mathcal{D}^j_{u,r}$ can be partitioned into r disjoint subsets:

$$\mathcal{D}^{j,l}_{u,r} = \Big\{ \mathbf{x} \in \Omega \,\Big|\, y_{j,I} = l \wedge \\ d(r) > C_j - D'(\mathbf{y}_j) \geq d(l) \Big\}, \qquad l \in \{0, \ldots, r-1\}.$$

If a state $\mathbf{x}$ belongs to the set $\mathcal{D}^{j,l}_{u,r}$, the state is a blocking state for link j, and the channel I is on layer l. Thus, the free capacity $C_j - D'(\mathbf{y}_j)$ of the link must be at most $d(r) - 1$, for the state to be a blocking state. The other channels may, however, consume at most $C_j - d(l)$ capacity units for the state to be within the allowed states. Now, let $v_j(l)$ denote the probability $\mathrm{P}\{\, \mathbf{X} \in \mathcal{D}^{j,l}_{u,r} \,\}$:

$$v_j(l) = p_{j,I}(l) \sum_{i=C_j-d(r)+1}^{C_j-d(l)} q_{j,I-1}(i),$$

where $q_{j,i}(x) = \mathrm{P}\{ S_{j,i} = x \}$. Here, $p_{j,i}(y) = \mathrm{P}\{ Y_{j,i} = y \}$, and is calculated by convolution, as shown in the next unnumbered subsection. The probability mass v_j of the set $\mathcal{D}^j_{u,r}$, can be calculated as

$$v_j = \mathrm{P}\{ \mathbf{X} \in \mathcal{D}^j_{u,r} \} = \sum_{l=0}^{r-1} v_j(l).$$

The link occupancy distribution $q_{j,I-1}(\cdot)$ can be calculated recursively by convolution:

$$q_{j,i}(x) = \sum_{y=0}^{x} q_{j,i-1}(x - d(y)) p_{j,i}(y),$$

where the recursion starts with $q_{j,0}(x) = 1_{x=0}$.

For interpretation of the convolution step, note that the event $\{S_{j,i} = x\}$ is the union of the events $\{Y_{j,i} = y, S_{j,i-1} = x - d(y)\}, y \in \{0, \ldots, L\}$. The corresponding probability is $q_{j,i-1}(x - d(y))p_{j,i}(y)$. Conversely, the conditional probability of the event $\{Y_{j,i} = y,\, S_{j,i-1} = x - d(y)\}$ given that $S_{j,i} = x$ is,

$$\mathrm{P}\{Y_{j,i} = y,\, S_{j,i-1} = x - d(y) \,|\, S_{j,i} = x\} = \frac{p_{j,i}(y) q_{j,i-1}(x - d(y))}{q_{j,i}(x)}. \tag{5.3}$$

Generating a sample state in $\mathcal{D}^j_{u,r}$ starts by drawing a value l for $Y_{j,I}$ using the distribution

$$\mathrm{P}\{Y_{j,I} = l \,|\, \mathbf{X} \in \mathcal{D}^j_{u,r}\} = \frac{\mathrm{P}\{ Y_{j,I} = l,\, \mathbf{X} \in \mathcal{D}^j_{u,r} \}}{\mathrm{P}\{ \mathbf{X} \in \mathcal{D}^j_{u,r} \}} = \frac{v_j(l)}{v_j},$$

where $l \in \{0, \ldots, r-1\}$.

Then, a value for $S'_j = S_{j,I-1}$ is drawn with the condition that $Y_{j,I} = l$ that is, using the distribution

$$p(x|l) = \mathrm{P}\{S_{j,I-1} = x \,|\, Y_{j,I} = l,\, \mathbf{X} \in \mathcal{D}^j_{u,r}\} = \frac{\mathrm{P}\{ Y_{j,I} = l, S_{j,I-1} = x \}}{\mathrm{P}\{ Y_{j,I} = l,\, \mathbf{X} \in \mathcal{D}^j_{u,r} \}},$$

since $\{Y_{j,I} = l \wedge S_{j,I-1} = x\} \subset \{\mathbf{X} \in \mathcal{D}^j_{u,r}\}$, restricting x to $x \in \{C_j - d(r) + 1, \ldots, C_j - d(l)\}$, and then

$$p(x|l) = \frac{p_{j,I}(l) q_{j,I-1}(x)}{v_j(l)} = \frac{q_{j,I-1}(x)}{\sum_{y=C_j-d(r)+1}^{C_j-d(l)} q_{j,I-1}(y)}.$$

Then, 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 probabilities in Equation (5.3). 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. Note that for reasonable sizes of links, it is advantageous to pre-calculate and to store the probabilities $p(x|l)$, $v_j(l)$ and v_j for fast generation of samples.

The next subsection presents a method for drawing leaf link states $\mathbf{Y}_u$, given the state $\mathbf{Y}_j$ of link j.

Generating leaf link states from a 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 $\mathbf{Y}_j = \max_{u \in \mathcal{U}_j}(\mathbf{Y}_u)$ using a similar inverse convolution procedure as above. Due to the assumed independence of channels, this condition can be broken down into separate conditions, i.e. for each i there is a separate problem of generating the values $Y_{u,i}$, $u \in \mathcal{U}$, under the condition $Y_{j,i} = \max_{u \in \mathcal{U}_j}(Y_{u,i})$ with a given $Y_{j,i}$. The above conditions affect leaf links $u \in \mathcal{U}_j$. For other links $u \in \mathcal{U} - \mathcal{U}_j$, the states $\mathbf{Y}_u$ are independently generated from the distribution $\pi_u(\cdot)$.

First, let us consider 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. This section uses an index $u_j \in \{1, \ldots, U_j\} = \mathcal{U}_j$ for the subset of leaf links. Let $Z_{u_j,i} = x$ denote the event that the channel is on state x on link j when leaf links $u' = 1, \ldots, u_j$ have been counted for, i.e.

$$Z_{u_j,i} = \max_{u' \leq u_j}(Y_{u',i}).$$

Note that $Y_{j,i} = Z_{U_j,i}$. Probabilities $\xi_{u_j,i}(s) = \mathrm{P}\{\, Z_{u_j,i} = s \,\}$ can be calculated recursively as follows:

$$\xi_{u_j,i}(s) = p_{u_j,i}(s) \sum_{s'=0}^{s-1} \xi_{u_j-1,i}(s') + \xi_{u_j-1,i}(s) \sum_{s'=0}^{s} p_{u_j,i}(s').$$

The recursion starts with $\xi_{0,i}(s) = 1_{s=0}$. The probabilities $p_{j,i}(s)$ used in the previous section are then simply $p_{j,i}(s) = \xi_{U_j,i}(s)$ where all users have been taken into account. If $Z_{u_j-1,i} = s$, then necessarily $Z_{u_j,i} \geq s$ (due to the nature of max-operation).

Conversely, to generate the state for each leaf link, given the value of $Y_{j,i}$, generate first $Z_{u_{j-1},i}$ from the distribution:

$$\mathrm{P}\{Z_{u_j-1,i} = x \,|\, Z_{u_j,i} = s\} = \begin{cases} \dfrac{\xi_{u_j-1,i}(x) \sum_{s'=0}^{x} p_{u_j,i}(s')}{\xi_{u_j,i}(s)}, & \text{when } x = s, \\ \dfrac{\xi_{u_j-1,i}(x) p_{u_j,i}(s)}{\xi_{u_j,i}(s)}, & \text{otherwise.} \end{cases}$$

Note that the event $Z_{u_j-1,i} < Z_{u_j,i}$ implies directly that $Y_{u_j,i} = Z_{u_j,i}$. If this is not the case, the value of $Y_{u_j,i}$ is drawn from the distribution

$$\mathrm{P}\{Y_{u_j,i} = y \,|\, Z_{u_j-1,i} = Z_{u_j,i} = s\} = \frac{p_{u_j,i}(y)}{\sum_{y'=0}^{s} p_{u_j,i}(y')}.$$

This procedure is repeated for each channel, resulting the state vectors of each leaf link $u \in \mathcal{U}_j$. The rest of the leaf link states are generated using the normal Monte Carlo simulation with distribution $\pi_u(\cdot)$.

Complexity

Generation of the link state in $\mathcal{D}_u^j$ takes approximately $R + IL$ steps, and is approximately as fast as generating a standard link state. Generating leaf link states from this link state takes $2IL(U-1)$ steps at maximum. Thus, compared to the standard Monte-Carlo method, generating samples with Inverse convolution takes at most twice the same time. 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 $R+L+I(C_j+2LU)$. It should be noted that the dependence on I and U is only linear, in spite of the exponential growth in I of the state space Ω.

The method scales very favourably with the network size, essentially defined by the number of user populations U. Apparently, the decomposition leads to replication of the simulation for each link on the route of the connection. The number of links on the route, however, grows relatively slowly, and in practice will never be very large.

Numerical examples

The publications present some numerical examples in order to illustrate the efficiency of the presented methods in Monte Carlo simulation of the blocking probabilities. The same network is used as in Nyberg *et al.* [58, 57], for which the exact blocking probability figures are known. The network is shown in Figure 5.2.

As can be seen, the variance reductions obtained with the inverse convolution method are remarkable both in the cases where multicast connections are layered and non-layered. The gain given by the inverse convolution method is better with light loads, which is natural, since the method is a typical application of the rare-event simulation techniques. In high load situations, however, the overhead in sample generation might not be justified, as the traditional Monte Carlo method gives rather good estimates, too.

5.3 User population models

Most of the publications discuss time blocking probabilities. To get the desired call blocking probabilities for the users, the user models need to be established. Publications [4] and [6] discuss several user population models, in a similar way as Nyberg *et al.* [57]. The “interface” between the user population models and the presented algorithms consists of two components. First, the user populations are assumed to generate a time reversible Markov process. Second, the parameter given to the algorithms, $\pi_u(\phi)$.

For the algorithms to work, detailed balance is required. The following subsections present user population models that satisfy the detailed balance condition, and give expressions for the corresponding $\pi_u(\phi)$ probabilities. The detailed balance requirement is naturally restrictive, but nevertheless, there are some useful user models that satisfy the requirement, as seen in this section.

Infinite user population

In an infinite user population model, calls to channels are generated as a Poisson arrival stream with arrival intensity λ_u, and a random sampling is done to assign the calls to different channels and layers. Call holding times (for all channels and layers) are assumed to be exponentially distributed with mean $1/\mu$. Channels and layers are selected with probabilities $\alpha_{u,i,l}$ ($\sum_{i\in\mathcal{I}}\sum_{l\in\mathcal{L}}\alpha_{u,i,l}=1$). Blocked calls are lost. The resulting traffic process is easily found to be a reversible Markov process. Since states of channels i on any leaf link u are independent,

$$\pi_u(\mathbf{y})=\prod_{i\in\mathcal{I}}p_{u,i,y_i},$$

where $p_{u,i,y_i}=\mathrm{P}\{\,Y_{u,i}=y_i\,\}$ is the probability that channel i is on layer y_i on leaf link u.

The probabilities $p_{u,i,l}$ are found by examining the probability of state 0 in an $M/M/\infty$ system (Publication [6]):

$$p_{u,i,0}=1-\sum_{l=1}^{L}p_{u,i,l},$$

$$p_{u,i,L}=1-e^{-\frac{\alpha_{u,i,L}\lambda_u}{\mu}},$$

and

$$p_{u,i,l}=\left(1-e^{-\frac{\alpha_{u,i,l}\lambda_u}{\mu}}\right)\prod_{m=l+1}^{L}e^{-\frac{\alpha_{u,i,m}\lambda_u}{\mu}}.$$

In the case of statistically indistinguishable channels, the equations for $\tilde{p}_{u,l}=p_{u,i,l}$ for all i are similar, except that $\alpha_{u,l}/I$ is substituted for $\alpha_{u,i,l}$, where $\sum_{l\in\mathcal{L}}\alpha_{u,l}=1$. Thus,

$$\pi_u(\mathbf{k})=\binom{I}{k_1\ \cdots\ k_L}\tilde{p}_{u,0}^{\,I-\sum_{l\in\mathcal{L}}k_l}\prod_{l\in\mathcal{L}}\tilde{p}_{u,l}^{k_l}.$$

Due to the PASTA property of traffic generated by the infinite user population, the call blocking probability equals the time blocking probability.

To study the insensitivity of the infinite user population model, the clocks associated with each state need to be defined (See Section 2.3). In order to do so, all the state diagrams of the leaf links are extended. In the extended state diagram of leaf link u, there are separate states corresponding to different numbers of actual users on each channel i and layer l. This results in a countably infinite number of states. If the call interarrival and connection holding times are exponentially distributed, the resulting process is a reversible Markov process.

Let $\mathcal{E}$ denote the extended state space. Now, for state vectors $\mathbf{n}\in\mathcal{E}$, and $\mathbf{n}+\mathbf{e}_{u,i,l}\in\mathcal{E}$, and state probabilities $\tilde{\pi}(\cdot)$, a detailed balance equation holds:

$$\tilde{\pi}(\mathbf{n})\lambda_u\alpha_{u,i,l}=\tilde{\pi}(\mathbf{n}+\mathbf{e}_{u,i,l})(n_{u,i,l}+1)\mu.$$

For general connection holding times, a scheme with relabelling for this associated Markov chain (see [70, Theorem 3.1]) is used. Associate each channel-layer pair with a clock $s_{u,i,l}$ that generates calls to that channel-layer pair. These clocks are active in all states of the system. In

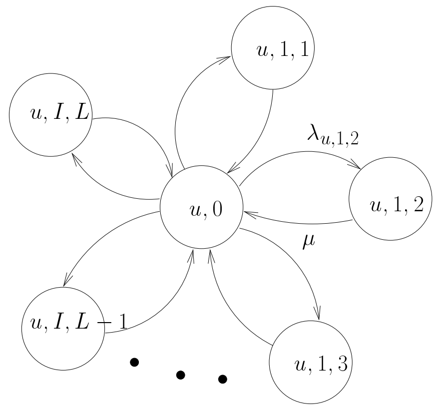

Figure 5.3: State transition diagram for user u (the single user model)

addition, each channel-layer pair is associated with clocks that represent the actual users on a specific channel-layer pair. For the insensitivity property to hold, the steady-state intensity of transitions due to birth of $s_{u,i,l,m}$ and due to death of $s_{u,i,l,m}$ must equal for all $m = 1, \ldots, n_{u,i,l}$. This can be seen to hold from the detailed balance equation of the extended state space. Thus, the stationary distribution of the system is insensitive to the call holding times.

Single user model

Consider the case where the user population consists of a single user, selecting each channel and layer with intensity $\lambda_{u,i,l}$ and holding them with intensity μ. Both user idle and call holding times are assumed to be exponentially distributed. The resulting Markov process is reversible. Note that the assumed independence between channels when using the inverse convolution algorithm restricts use of this user population model to the case where users cannot select between channels, but are always either off or on a specific channel. Figure 5.3 shows the state transition diagram of the process.

The probability that the user is on channel i and layer l is

$$p_{u,i,l} = \frac{\lambda_{u,i,l}/\mu}{1 + \sum_{i' \in \mathcal{I}} \sum_{l' \in \mathcal{L}} \lambda_{u,i',l'}/\mu},$$

and the probability that the user is idle is

$$p_{u,0} = \frac{1}{1 + \sum_{i' \in \mathcal{I}} \sum_{l' \in \mathcal{L}} \lambda_{u,i',l'}/\mu}.$$

The probabilities $\pi_u(\mathbf{y})$ can be calculated from these probabilities as

$$\pi_u(\mathbf{y}) = \begin{cases} p_{u,0} & \text{when } \mathbf{y} = \mathbf{0}, \\ p_{u,i,l} & \text{when } \mathbf{y} = l\mathbf{e}_i, \\ 0 & \text{otherwise.} \end{cases}$$

For the case of statistically indistinguishable channels, use $\lambda_{u,l}/I$ instead of $\lambda_{u,i,l}$. The probabilities $\pi_u(\mathbf{k})$ are then

$$\pi_u(\mathbf{k}) = \begin{cases} p_{u,0} & \text{when } \mathbf{y} = \mathbf{0}, \\ \sum_{i \in \mathcal{I}} p_{u,i,l} & \text{when } \mathbf{k} = \mathbf{e}_l, \\ 0 & \text{otherwise.} \end{cases}$$

With the single user model, call blocking equals time blocking in a network without this user, as discussed in Section 2.4. Thus, for calculating call blocking for a specific user, it is just necessary to use for this user $p_{u,0} = 1$ and $p_{u,i,l} = 0$ for $l > 0$ and for all i.

To study the insensitivity properties, each user u is associated with a set of clocks (see Section 2.3): clock $s_{u,0}$ for the state where the user is not on any channel or layer, and clocks $s_{u,i,l}$ for each channel-layer pair. In each state, only one of these clocks can be on. If the user is on a specific channel i and layer l, $s_{u,i,l}$ is on. If the user is not on any channel, clock $s_{u,0}$ is on. The resulting state space is locally balanced with respect to all clocks $s \in \mathcal{T}$. Thus, the steady state distribution of the system is insensitive to both call holding times and user idle times, but the call blocking probabilities are insensitive only to call holding times.

Finite user population of independent users

Finite user populations with independent users can be generated using the presented single user model. Just generate a tree with supplemented links of infinite capacity, where each leaf node contains a single user, and the root link of the tree is the leaf link of the original problem. Although the approach is simple, in some cases the generated tree might be prohibitively large and require excessive computation.

Consider, for example, a user population consisting of a finite number of independent layer-specific users. Each user is defined as a single user, as in the previous section, who only visits the states of a specific layer and the idle state.

Call blocking with finite user populations is calculated by first removing the user we are calculating blocking for (by setting $p_{u,0} = 1$), and then calculating the time blocking probability of the network.

The system with a finite user population has the insensitivity properties of the actual users used to construct the user population.

6 SIGNALLING LOAD FOR MULTICAST

Publication [3] studies the signalling loads generated by dynamic multicast trees. The analysis is restricted to the load generated by arriving and leaving users. The analysis is based on counting the number of events per time unit (rate) that require call establishments or teardowns. The processing load incurred in the nodes can be calculated from these rates.

The model used is more general than the multicast loss system, and assumes that the network is non-blocking. Thus, for real networks where blocking occurs, the analysis gives only approximate (but conservative) results.

6.1 Signalling action rates in links

There are two types of signalling actions: (1) actions that establish or tear down connections, and (2) actions that do not change the set of connections established in the link. This section presents a method for calculating the average signalling action rates of the first type in links for multicast trees with a finite and heterogeneous user population. In the next section these rates are used to calculate the processing costs at nodes for different types of signalling protocols. The signalling action rates of the second type are outside the scope of this dissertation. They depend on the signalling task, and can easily be calculated using similar procedures as for the first type.

Consider a network with a general topology, i.e. an undirected graph. The network has a routing algorithm that gives the legs for each possible multicast tree. A multicast tree is a tree-like connection that is established according to the multicast routing scheme. The multicast tree may be dynamic, i.e. it has different receivers at different moments, but the underlying routing is assumed to be static.

Let I denote the set of multicast trees in this network. Let each tree carry only one channel i. Consider then an arbitrary multicast tree $i \in I$. Let $\mathcal{U}_i$ denote the set of leaves of this multicast tree. Note that each leaf $u \in \mathcal{U}_i$ corresponds to a potential user of multicast tree i. Each user $u \in \mathcal{U}_i$ behaves independently of each other. Further, the users are of the on/off-type with respect to the multicast tree. More precisely, the active periods $T_{i,u}^{\mathrm{on}}$ (as well as the idle periods $T_{i,u}^{\mathrm{off}}$) are independent and identically distributed with a general (user-dependent) distribution. Thus, user u establishes connections to the multicast tree at rate $\lambda_{i,u} > 0$, and the probability that this user is connected to the tree (at an arbitrary time point) is $p_{i,u}$, where

$$\lambda_{i,u} = \frac{1}{\mathrm{E}[T_{i,u}^{\mathrm{on}}] + \mathrm{E}[T_{i,u}^{\mathrm{off}}]}, \qquad \text{and} \qquad p_{i,u} = \frac{\mathrm{E}[T_{i,u}^{\mathrm{on}}]}{\mathrm{E}[T_{i,u}^{\mathrm{on}}] + \mathrm{E}[T_{i,u}^{\mathrm{off}}]}, \quad (6.1)$$

which follows from the assumption of infinite capacity links.

Consider then an arbitrary node j of the network. The link upstream from node j (i.e. towards the root) is called link j (with respect to multicast tree i). Note that the same link may have a different index in another multicast tree. A subtree rooted at node j is called subtree j. The set of all users in this subtree is denoted by $\mathcal{U}_{i,j}$. If the multicast connection is already carried in the link when a downstream user connects to the tree, no action is made in the link. Similarly, if any other downstream user is connected to the multicast tree, no action is taken when a user downstream disconnects from the tree. The probability of any of the other users being connected to the tree is the same in both cases. Due to this symmetry, the rest of the analysis considers only connection establishment.

Consider connection establishments for multicast tree i arriving to node j. User $u \in \mathcal{U}_{i,j}$ tries to establish this type of connections at rate $\lambda_{i,u}$. However, such a connection establishment will affect link j only if the multicast connection is not already carried in that link, i.e. all the other users $v \in \mathcal{U}_{i,j} \setminus \{u\}$ are idle, which will happen with probability $P_{i,j} = \prod_{v \in \mathcal{U}_{i,j} \setminus \{u\}} (1 - p_{i,v})$ because of the independence assumption concerning the users. The actual arrival rate is $\lambda_{i,u} P_{i,j}$.

Summation over all the users $u \in \mathcal{U}_{i,j}$ yields the rate $R_{i,j}$ of connection establishments (corresponding to multicast tree i) in link j:

$$R_{i,j} = \sum_{u \in \mathcal{U}_{i,j}} \lambda_{i,u} \prod_{v \in \mathcal{U}_{i,j} \setminus \{u\}} (1 - p_{i,v}).$$

Note that, due to symmetry mentioned above, the equation yields the connection teardown rate in link j as well.

If the user population $\mathcal{U}_i$ is homogeneous, $\lambda_{i,u} = \lambda_i$ and $p_{i,u} = p_i$ for all $u \in \mathcal{U}_i$ implying that

$$R_{i,j} = U_{i,j} \lambda_i (1 - p_i)^{U_{i,j} - 1}, \tag{6.2}$$

where $U_{i,j}$ is the size of set $\mathcal{U}_{i,j}$.

6.2 Average processing costs at nodes

The signalling action rate presented in the previous section is a link-oriented figure. This section considers the required processing cost for signalling actions in network nodes. The processing costs depend on the signalling protocol used. Publication [3] presents a method for calculating average processing costs (per time unit) for three types of signalling protocols, namely (1) Network Leaf Initiated Join, (2) Root Leaf Initiated Join and (3) a protocol with a separate group manager node controlling joins. The protocols are illustrated in figure 6.1.

Let $\mathcal{N}_{i,j}$ denote the downstream neighbour links of the multicast tree i rooted in node j. The rate of incoming connection establishment (or teardown) actions to node j and the rate of outgoing connection establishments (or teardowns) in node j are as follows:

$$R_j^{\text{in}} = \sum_{i \in I} \sum_{b \in \mathcal{N}_{i,j}} R_{i,b} \qquad \text{and} \qquad R_j^{\text{out}} = \sum_{i \in I} R_{i,j}. \tag{6.3}$$

The total rates of incoming and outgoing connection establishment (or teardown) requests given by users downstream of node j are

$$\Lambda_j^{\text{in}} = \sum_{i \in I} \sum_{b \in \mathcal{N}_{i,j}} \sum_{u \in \mathcal{U}_{i,b}} \lambda_{i,u} \qquad \text{and} \qquad \Lambda_j^{\text{out}} = \sum_{i \in I} \sum_{u \in \mathcal{U}_{i,j}} \lambda_{i,u} \tag{6.4}$$

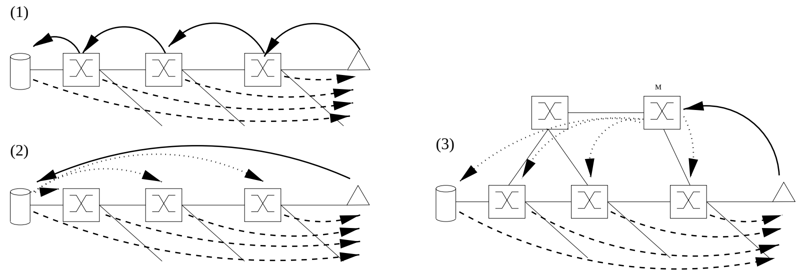

Figure 6.1: Signalling in different protocols. (1) each node accepts new connections independently. (2) Root node controls joining and leaving the tree. (3) node M controls joining and leaving of the tree. '—' connection requests, '$\cdots$' connection establishment permissions, and '– –' connection establishments.

respectively. Since $\mathcal{U}_{i,j} = \bigcup_{b \in \mathcal{N}_{i,j}} \mathcal{U}_{i,b}$, $\Lambda_j^{\text{in}} = \Lambda_j^{\text{out}} = \Lambda_j$.

Let c^{est} denote the processing cost of a single node for establishing a connection in a link. Note that the nodes at both ends of the link are affected. Similarly, c^{fwd} denotes the cost of forwarding a signalling request via a link and c^{trd} the cost of a connection teardown for a link. c^{adm} denotes the cost of making the decision to admit a new member to join the multicast tree.

Protocol 1. Network Leaf Initiated Join is a signalling protocol in which a leaf node that wants to join the multicast tree sends a join request upstream. When a node that is already connected to the tree is encountered, the connected node makes the decision to accept the new connection and establishes the new branch to the tree.

The processing cost for a network node in this scenario consists of the cost of forwarding the join requests upstream $C_j^{\text{fwd}} = c^{\text{fwd}}(R_j^{\text{in}} + R_j^{\text{out}})$, the cost of connections to be established for down- and upstream, $C_j^{\text{est}} = c^{\text{est}}(R_j^{\text{in}} + R_j^{\text{out}})$, the cost of connection teardowns, $C_j^{\text{trd}} = c^{\text{trd}}(R_j^{\text{in}} + R_j^{\text{out}})$, and the cost of connection admissions, $C_j^{\text{adm}} = c^{\text{adm}}(R_j^{\text{in}} - R_j^{\text{out}})$. The total cost for node j is $C_j = C_j^{\text{est}} + C_j^{\text{fwd}} + C_j^{\text{trd}} + C_j^{\text{adm}}$.

Protocol 2. Root Leaf Initiated Join is a signalling protocol in which a leaf node that wants to join the multicast tree sends a join message to the root of the tree. The root of the multicast tree makes the decision to accept the new connection.

The cost of forwarding the join requests upstream is in this scenario $C_j^{\text{fwd}} = c^{\text{fwd}}(\Lambda_j^{\text{in}} + \Lambda_j^{\text{out}}) = 2c^{\text{fwd}}\Lambda_j$, and the cost of forwarding the permission to accept the new connection downstream, which only affects links in which the corresponding connection is not established, is $C_j^{\text{fwd-d}} = c^{\text{fwd}}(2\Lambda_j - R_j^{\text{in}} - R_j^{\text{out}})$. The costs of connections establishments C_j^{est} and teardowns C_j^{trd} for down- and upstream are the same as for protocol 1. The cost of connection admission is nonzero only for the root node, $C_j^{\text{adm}} = c^{\text{adm}}\Lambda_j 1_{j=\text{root}}$, where $1_{(\cdot)}$ is the indicator function.

The total cost for node j is $C_j = C_j^{\mathrm{est}} + C_j^{\mathrm{fwd}} + C_j^{\mathrm{fwd-d}} + C_j^{\mathrm{trd}} + C_j^{\mathrm{adm}}$. Note that, since $2\Lambda_j \geq R_j^{\mathrm{in}} + R_j^{\mathrm{out}}$, the processing cost without the cost of connection admission for protocol 2 is always greater than that for protocol 1.

Protocol 3. A protocol with a separate group manager is a signalling protocol in which a leaf node that wants to join the multicast tree sends a join message to the group manager node. The group manager node then accepts the connection and performs signalling to establish the new branch to the multicast tree. This case is handled by separating actions in two trees: the signalling tree which carries the establishment requests and the actual multicast tree.

To handle the effect of dynamic routing decisions, the signalling rates for each link may be random sampled with the probabilities of the routing functions to choose the corresponding paths.

6.3 Application: TV distribution network

Publication [3] studies a TV distribution network as an application. The network is implemented as a circuit switched telecommunications network. The signalling model used is Network Leaf Initiated Join (protocol 1), to avoid a large number of receivers creating a bottleneck to a centralized group management node.

It is reasonable to assume that the topology of the underlying (distribution) network is a tree. In the root of this distribution tree, a server offers the users in the leaves a set I of programmes, or channels, which are running independently of the users. Each channel $i \in I$ sets up a dynamic multicast subtree with $\mathcal{U}_i = \mathcal{U}$, where $\mathcal{U}$ denotes the set of all leaves in the underlying distribution tree. That is, the multicast loss system with infinite link capacities is used.

The users $u \in \mathcal{U}$ of the TV distribution network are assumed to be homogeneous. They are characterized by a preference distribution p_i, $i \in \mathcal{I}$, where p_i is the probability that the user is connected to channel i at an arbitrary time point. At any time, each user is connected to exactly one channel. Thus, $\sum_{i \in \mathcal{I}} p_i = 1$.

The preference distribution, however, does not tell anything about the rate at which new connection establishments arrive. Therefore, a "channel surfing" model is needed. The publication uses a simple single user model where the user is always on some channel, and changes the channel at rate λ^{tot}. The new channel is drawn (independently of the former one) from the preference distribution p_i. Thus, the connection establishment rate λ_i for channel i is $\lambda_i = \lambda^{\mathrm{tot}} p_i$. Note that the parameters p_i and λ^{tot} determine the mean lengths of the active and idle periods corresponding to channel i; equation (6.1) yields

$$\mathrm{E}[T_i^{\mathrm{on}}] = \frac{1}{\lambda^{\mathrm{tot}}} \qquad \text{and} \qquad \mathrm{E}[T_i^{\mathrm{off}}] = \frac{1-p_i}{p_i}\,\frac{1}{\lambda^{\mathrm{tot}}}.$$

Consider now an arbitrary link j of the distribution tree. Note that in this case the index of a link is the same for all multicast trees. Let $\mathcal{U}_j$ denote the set of all leaves in subtree j. Under the assumptions made above, it follows from (6.2), that the connection establishment rate for channel i in link j is $R_{i,j} = \lambda^{\text{tot}} U_j p_i (1-p_i)^{U_j-1}$. Summing over the channels $i \in I$ yields the total connection establishment rate R_j in link j:

$$R_j = \sum_{i \in \mathcal{I}} R_{i,j} = \lambda^{\text{tot}} U_j \sum_{i \in \mathcal{I}} p_i (1-p_i)^{U_j-1}.$$

Remind that for the chosen signalling model (Protocol 1) the processing cost C_j due to signalling load at node j depends linearly both on the sum of the incoming and outgoing connection establishment rates, $R_j^{\text{in}} + R_j^{\text{out}}$, and their difference, $R_j^{\text{in}} - R_j^{\text{out}}$:

$$C_j = (c^{\text{fwd}} + c^{\text{est}} + c^{\text{trd}})(R_j^{\text{in}} + R_j^{\text{out}}) + c^{\text{adm}}(R_j^{\text{in}} - R_j^{\text{out}}). \qquad (6.5)$$

These rates are based on the connection establishment rates R_j in links as follows: $R_j^{\text{in}} = \sum_{b \in \mathcal{N}_j} R_b$ and $R_j^{\text{out}} = R_j$, where $\mathcal{N}_j$ refers to the set of downstream neighbour links of node j.

The publication demonstrates the effect of the structure of the network and the users' preference distribution on the signalling load, by calculating the processing costs for some example networks. The results show that both the connection establishment rates and the processing costs depend considerably on the users' preference distribution. Thus, as regards the signalling load at nodes, the uniform distribution is *not* the worst case as one might imagine. In all, modifying the structure had a positive effect making the signalling load more even at various levels.

7 KEY RESULTS AND AUTHOR'S CONTRIBUTION

7.1 Summary

This dissertation presents new algorithms that can be used when estimating grade of service in networks with dynamic multicast connections. The algorithms use techniques derived from the probability and queueing theory to get grade of service figures, specifically the call blocking probabilities experienced by the users of the system.

The analysis is mainly concentrated on tree type networks, where algorithms and simulation methods for solving the blocking probabilities exactly are derived. Both single layer and hierarchically coded streams are treated. The presented algorithms reduce significantly the computational complexity of the problem, compared to direct calculation from the system state space. An approximative method is given for the inclusion of background traffic.

The simulation method presented is an application of the Inverse Convolution Monte-Carlo method, and it gives a considerable variance reduction, and thus allows simulation with smaller sample sizes than with traditional simulation methods.

Blocking probabilities are also studied in a single link setting, where an algorithm for call blocking probabilities is derived, and applied for approximative analysis of end-to-end blocking probabilities in networks. Blocking probabilities in a cellular system are studied by means of simulation.

Signalling load caused by the users joining and leaving the multicast tree is examined. A method for calculating the rates at which these signalling events occur is presented, and three types of signalling protocols involved are studied.

7.2 Contribution of this dissertation

This section describes the contributions of each publication in this dissertation, and states the role of the author.

Publication [1] presents a model for calculating blocking probabilities of multicast connections in a single link. The author has written Sections 1, 4 and 5. Sections 2 and 3 are joint work with co-authors.

Publication [2] presents an application of the Reduced Load Approximation to the multicast case. The paper is strongly based on Publication [1]. Sections 3 and 4 are mainly written by the author, but are joint work with the other authors. The simulation scenarios are joint work, but the simulations were carried out by the author.

Publication [3] studies how the signalling load due to multicast connections is distributed in the network. The work is mainly author's, but Section 4 is Dr Aalto's work.

Publication [4] presents an extension to the publication of Aalto and Virtamo [10] for multiple groups of statistically indistinguishable channels. The author's contribution is mainly in Section 4 for generalisation of the earlier work, made together with the authors of the original work, and in Section 5 that is author's work.

Publication [5] generalises the earlier results of Nyberg *et al.* [58] to the case where multicast transmissions use two-layer hierarchical coding. The author extended the required state-space definition. The required extension to combinatorial convolution was joint work by the author and the co-authors. Sections 3.2, 3.3 and 4 are author's work.

Publication [6] generalises the results of Publication [5] to the case of multiple layers. The author brought the concept of supplemented trees to the publication. In addition, Sections 3 and 4 are mostly of the author's work.

Publication [7] applies the results of Lassila and Virtamo [51] to the multicast setting. The author wrote most of Sections I, II, IIA, IV, IVA, IVB, IVC. The author conducted all simulations with the simulation program coded together with Dr Lassila.

Publication [8] was written solely by the author and generalises the results of Publication [7] to the case with multi-layer multicast streams.

Publication [9] studies the multicast gain over unicast in terms of the number of supported users in a cell of a cellular network. Most of the work is a joint effort of all the authors. The author programmed the simulation programs and conducted all the simulations.

Proof of Lemma 4.2 of Publication [6]

Due to space restrictions, proof of Lemma 4.2 was left out from Publication [6]. It is provided here for completeness.

First, assume that link j has two downstream neighbour links, $\mathcal{N}_j = \{s,t\}$. The set of channels that are on layer l on link s is denoted by $\mathcal{L}_l \subset \mathcal{I}$. These sets are disjoint, i.e. for all $l \neq l'$, $\mathcal{L}_l \cap \mathcal{L}_{l'} = \emptyset$. The set of channels that are on layer l on link t is denoted by $\mathcal{M}_l \subset \mathcal{I}$. These sets are also disjoint; for all $l \neq l'$, $\mathcal{M}_l \cap \mathcal{M}_{l'} = \emptyset$. The intersections of these sets are denoted as $\mathcal{L}_1 \cap \mathcal{M}_1 = \mathcal{X}_{1,1}$, $\mathcal{L}_1 \cap \mathcal{M}_2 = \mathcal{X}_{1,2}$, $\mathcal{L}_2 \cap \mathcal{M}_1 = \mathcal{X}_{2,1}$, $\mathcal{L}_2 \cap \mathcal{M}_2 = \mathcal{Y}_{2,2}$, and so on. For a two layer example, see Figure 7.1. The set of channels that are on layer l on link j is denoted by $\mathcal{K}_l$. Let $l_i = |\mathcal{L}_i|$, $m_i = |\mathcal{M}_i|$, and $k_i = |\mathcal{K}_i|$ for all $i = 1,\ldots,L$ Let $x_{a,b} = |\mathcal{X}_{a,b}|$ for all $a, b \in \mathcal{L}$. Let $\mathbf{l} = (l_1,\ldots,l_L)$, $\mathbf{m} = (m_1,\ldots,m_L)$, $\mathbf{k} = (k_1,\ldots,k_L)$.

The following notation is used here for the multinomial coefficient:

$$\binom{a}{b_1 \ \cdots \ b_n} = \frac{a!}{(a - \sum_{n'=1}^{n} b_{n'})! \prod_{n'=1}^{n} (b_{n'}!)}.$$

In the general case, there are L^2 possible intersections for the sets. Of these, however, L intersections are defined by the others. The intersections $x_{a,a}$ are chosen as such:

$$x_{a,a} = l_a + m_a - k_a - \sum_{a'=a+1}^{L} \left(x_{a,a'} + x_{a',a}\right).$$

From now on, the short notation of style $\mathrm{P}\{\,\mathbf{k}\,\}$ is used for probabilities $\mathrm{P}\{\,\mathbf{K} = \mathbf{k}\,\}$. With these definitions, the probability $\mathrm{P}\{\mathbf{K}_j = \mathbf{k} \mid \mathbf{K}_s = \mathbf{l}, \mathbf{K}_t = \mathbf{m}\} = \mathrm{P}\{\mathbf{k}\,|\,\mathbf{l},\mathbf{m}\}$ is easily calculated from the following lemma.

Lemma 4.2 of Publication [6] *For three links j, s, and t, such that $\mathcal{N}_j = \{s,t\}$, carrying multicast traffic with L layers and I statistically indistinguishable channels, the following equality holds*

$$\mathrm{P}\{\mathbf{k}\,|\,\mathbf{l},\mathbf{m}\} = \\ = \sum_{x_{1,2}} \cdots \sum_{x_{1,L}} \sum_{x_{2,1}} \sum_{x_{2,3}} \cdots \sum_{x_{2,L}} \cdots \sum_{x_{L,1}} \cdots \sum_{x_{L,L-1}} \frac{\binom{I - \sum_{a \geq 1} l_a}{m_1 - \sum_{a\geq 1} x_{a,1} \ \cdots \ m_L - \sum_{a \geq 1} x_{a,L}}}{\binom{I}{m_1 \ \cdots \ m_L}} \prod_{a=1}^{L} \binom{l_a}{x_{a,1} \ \cdots \ x_{a,L}}.$$

Proof: – Step 1: Follows directly from the one-layer case (Publication [4])

– Induction step:

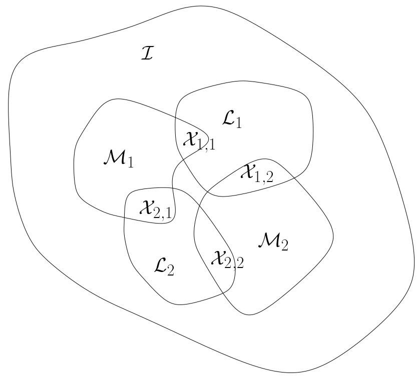

Figure 7.1: Combining two links when $L = 2$. All channels in $\mathcal{I}$ are equally probable. Link s has $|\mathcal{L}_1| = l_1$ channels on layer 1 and $|\mathcal{L}_2| = l_2$ channels on layer 2. Sets $\mathcal{L}_1$ and $\mathcal{L}_2$ are disjoint. Similarly, link t has m_1 and m_2 channels on layers 1 and 2, respectively.

First, rewrite the product inside $\mathrm{P}\{\,\mathbf{k}|\mathbf{l},\mathbf{m}\,\}$ as follows:

$$\frac{\binom{I-\sum_{a\geq 1} l_a}{m_1-\sum_{a\geq 1} x_{a,1}\ \cdots\ m_L-\sum_{a\geq 1} x_{a,L}}}{\binom{I}{m_1\ \cdots\ m_L}} \prod_{a=1}^{L} \binom{l_a}{x_{a,1}\ \cdots\ x_{a,L}} =$$

$$\overbrace{\prod_{a=1}^{L-1} \left\{ \frac{\binom{l_a}{x_{a,1}\ \cdots\ x_{a,L}} \binom{I-\sum_{b\geq a} l_b}{m_1-\sum_{b\geq a} x_{b,1}\ \cdots\ m_L-\sum_{b\geq a} x_{b,L}}}{\binom{I-\sum_{b>a} l_b}{m_1-\sum_{b>a} x_{b,1}\ \cdots\ m_L-\sum_{b>a} x_{b,L}}} \right\}}^{\text{Part A}}$$

$$\times \frac{\overbrace{\binom{l_L}{x_{L,1}\ \cdots\ x_{L,m_L}} \binom{I-l_L}{m_1-x_{L,1}\ \cdots\ m_L-x_{L,L}}}^{\text{Part B}}}{\binom{I}{m_1\ \cdots\ m_L}} .$$

Introduce a new, lower layer 1 to a system with $L-1$ layers having layer indices $\{2,\ldots,L\}$. Define $x_{1,1} = l_1 + m_1 - k_1 - \sum_{a=2}^{L}(x_{1,a} + x_{a,1})$.

Now (using a shorthand notation):

$$\begin{aligned}
\mathrm{P}\{\mathbf{k}\,|\,\mathbf{l},\mathbf{m}\} =& \mathrm{P}\{k_1,\ldots,k_L\,|\,\mathbf{l},\mathbf{m}\} \\
=& \sum_{x_{1,2}} \cdots \sum_{x_{1,L}} \sum_{x_{2,1}} \sum_{x_{2,3}} \cdots \sum_{x_{2,L}} \cdots \sum_{x_{L,1}} \cdots \sum_{x_{L,L-1}} \\
& \mathrm{P}\{x_{1,1},\ldots,x_{1,L} \cdots x_{L,1},\ldots,x_{L,L}\,|\,\mathbf{l},\mathbf{m}\}
\end{aligned}$$

$$= \sum_{x_{2,3}} \cdots \sum_{x_{2,L}} \cdots \sum_{x_{L,2}} \cdots \sum_{x_{L,L-1}} \underbrace{\mathrm{P}\{x_{2,2},\ldots,x_{2,L}\cdots x_{L,2},\ldots,x_{L,L}\,|\,\mathbf{l},\mathbf{m}\}}_{\text{Part 1}}$$

$$\times \sum_{x_{1,2}} \cdots \sum_{x_{1,L}} \sum_{x_{2,1}} \cdots \sum_{x_{L,1}} \Bigg\{ \underbrace{\mathrm{P}\{x_{1,1}\,|\,x_{1,2},\ldots,x_{1,L}\cdots x_{L,1},\ldots,x_{L,L},\mathbf{l},\mathbf{m}\}}_{\text{Part 2}}$$

$$\times \underbrace{\prod_{a=2}^{L-1} \mathrm{P}\{x_{1,a}\,|\,x_{1,a+1},\ldots,x_{1,L}\cdots x_{L,1},\ldots,x_{L,L},\mathbf{l},\mathbf{m}\}}_{\text{Part 3}}$$

$$\times \underbrace{\mathrm{P}\{x_{1,L}\,|\,x_{2,1},\ldots,x_{2,L}\cdots x_{L,1},\ldots,x_{L,L},\mathbf{l},\mathbf{m}\}}_{\text{Part 4}}$$

$$\times \underbrace{\prod_{a=2}^{L-1} \mathrm{P}\{x_{a,1}\,|\,x_{a+1,1},\ldots,x_{L,1}\cdots x_{2,L},\ldots,x_{L,L},\mathbf{l},\mathbf{m}\}}_{\text{Part 5}}$$

$$\times \underbrace{\mathrm{P}\{x_{L,1}\,|\,x_{2,2},\ldots,x_{2,L}\cdots x_{L,2},\ldots,x_{L,L},\mathbf{l},\mathbf{m}\}}_{\text{Part 6}} \Bigg\}$$

Consider Part 1, $\mathrm{P}\{x_{2,2},\ldots,x_{2,L}\cdots x_{L,2},\ldots,x_{L,L} \mid \mathbf{l},\mathbf{m}\}$. There is no remaining dependence on l_1 or m_1, thus this probability corresponds to the probability $\mathrm{P}\{\mathbf{k}'\,|\,\mathbf{l}',\mathbf{m}'\}$, i.e. the probability calculated without the added layer 1.

Part 6:

$$\mathrm{P}\{x_{L,1}\,|\,x_{2,2},\ldots,x_{2,L}\cdots x_{L,2},\ldots,x_{L,L},\mathbf{l},\mathbf{m}\} = \frac{\binom{l_L-\sum_{b=2}^{L} x_{L,b}}{x_{L,1}}\binom{I-\sum_{b=2}^{L}(m_b-x_{L,b})-l_L}{m_1-x_{L,1}}}{\binom{I-\sum_{b=2}^{L} m_a}{m_1}}.$$

Multiplying this probability by Part B of the probability $\mathrm{P}\{\mathbf{k}' \mid \mathbf{l}',\mathbf{m}'\}$ yields

$$\frac{\binom{l_L-\sum_{b=2}^{L} x_{L,b}}{x_{L,1}}\binom{I-\sum_{b=2}^{L}(m_b-x_{L,b})-l_L}{m_1-x_{L,1}}}{\binom{I-\sum_{b=2}^{L} m_a}{m_1}} \times \frac{\binom{l_L}{x_{L,2}\ \cdots\ x_{L,m_L}}\binom{I-l_L}{m_2-x_{L,2}\ \cdots\ m_L-x_{L,L}}}{\binom{I}{m_2\ \cdots\ m_L}} = \frac{\binom{l_L}{x_{L,1}\ \cdots\ x_{L,m_L}}\binom{I-l_L}{m_1-x_{L,1}\ \cdots\ m_L-x_{L,L}}}{\binom{I}{m_1\ \cdots\ m_L}},$$

i.e. Part B of probability $\mathrm{P}\{\mathbf{k}\,|\,\mathbf{l},\mathbf{m}\}$.

Correspondingly, Part 5 is as follows

$$\mathrm{P}\{x_{a,1}\,|\,x_{a+1,1},\ldots,x_{L,1}\cdots x_{2,L},\ldots,x_{L,L},\mathbf{l},\mathbf{m}\} = \frac{\binom{l_a-\sum_{b>1}x_{a,b}}{x_{a,1}}\binom{I-\sum_{b\geq a}l_b-\sum_{b>1}(m_b-\sum_{c\geq b}x_{c,b})}{m_1-\sum_{b\geq a}x_{b,1}}}{\binom{I-\sum_{b>a}l_b-\sum_{b>1}(m_b-\sum_{c>b}x_{c,b})}{m_1-\sum_{b>a}x_{b,1}}}.$$

Multiplying these probabilities with the terms with the same a in Part A of $\mathrm{P}\{\mathbf{k}'\,|\,\mathbf{l}',\mathbf{m}'\}$ yields

$$\prod_{a=2}^{L-1}\left\{\frac{\binom{l_a}{x_{a,1}\cdots x_{a,L}}\binom{I-\sum_{b\geq a}l_b}{m_1-\sum_{b\geq a}x_{b,1}\cdots m_L-\sum_{b\geq a}x_{b,L}}}{\binom{I-\sum_{b>a}l_b}{m_1-\sum_{b>a}x_{b,1}\cdots m_L-\sum_{b>a}x_{b,L}}}\right\},$$

and only the term with $a = 1$ is now missing. It will be the product of the remaining conditional probabilities.

Part 2:

$$\mathrm{P}\{x_{1,1}\,|\,x_{1,2},\ldots,x_{1,L}\cdots x_{L,1},\ldots,x_{L,L},\mathbf{l},\mathbf{m}\} = \frac{\binom{l_1-\sum_{b>1}x_{1,b}}{x_{1,1}}\binom{I-\sum_{b\geq 1}l_b-\sum_{b>1}(m_b-\sum_{c\geq 1}x_{c,b})}{m_1-\sum_{b\geq 1}x_{b,1}}}{\binom{I-\sum_{b>1}l_b-\sum_{b>1}(m_b-\sum_{c>1}x_{c,b})}{m_1-\sum_{b>1}x_{b,1}}}, \tag{7.1}$$

Part 3:

$$\mathrm{P}\{x_{1,a}\,|\,x_{1,a+1},\ldots,x_{1,L}\cdots x_{L,1},\ldots,x_{L,L},\mathbf{l},\mathbf{m}\} = \frac{\binom{l_1-\sum_{b>a}x_{1,b}}{x_{1,a}}\binom{I-\sum_{b\geq 1}l_b-\sum_{b>a}(m_b-\sum_{c\geq 1}x_{c,b})}{m_a-\sum_{b\geq 1}x_{b,a}}}{\binom{I-\sum_{b>1}l_b-\sum_{b>a}(m_b-\sum_{c>1}x_{c,b})}{m_a-\sum_{b>1}x_{b,a}}}, \tag{7.2}$$

and Part 4:

$$\mathrm{P}\{x_{1,L}\,|\,x_{2,1},\ldots,x_{2,L}\cdots x_{L,1},\ldots,x_{L,L},\mathbf{l},\mathbf{m}\} = \frac{\binom{l_1}{x_{1,L}}\binom{I-\sum_{b\geq 1}l_b}{m_L-\sum_{b\geq 1}x_{b,L}}}{\binom{I-\sum_{b>1}l_b}{m_L-\sum_{b>1}x_{b,L}}}. \tag{7.3}$$

Multiplying Equations (7.1) and (7.3) with the product of (7.2) for all $a = 2,\ldots,L-1$ yields

$$\frac{\binom{l_1}{x_{1,1}\cdots x_{1,L}}\binom{I-\sum_{b\geq 1}l_b}{m_1-\sum_{b\geq 1}x_{b,1}\cdots m_L-\sum_{b\geq 1}x_{b,L}}}{\binom{I-\sum_{b>1}l_b}{m_1-\sum_{b>1}x_{b,1}\cdots m_L-\sum_{b>1}x_{b,L}}},$$

i.e. the first factor of Part A of the probability $\mathrm{P}\{\mathbf{k}\,|\,\mathbf{l},\mathbf{m}\}$. □

APPENDIX II ERRATUM

1. Publication [1]:

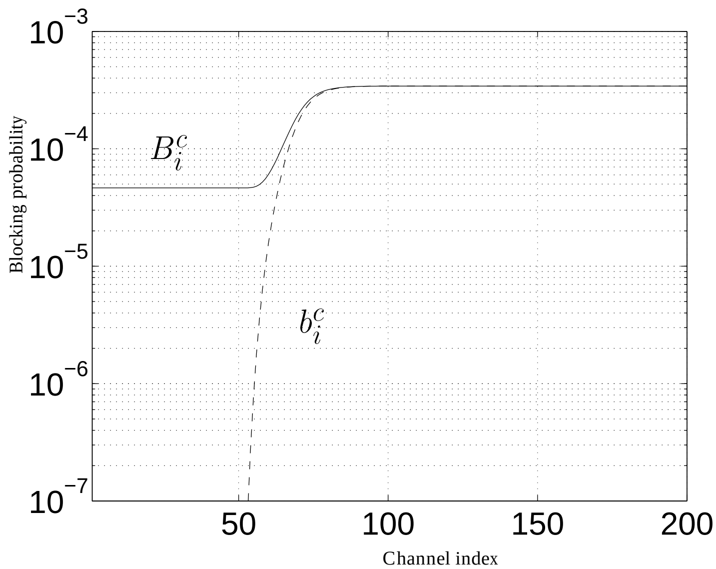

Figure 6 Channel B_i^c and call blocking probabilities b_i^c for each channel (B_i^c solid line, b_i^c dashed line).

BIBLIOGRAPHY

[1] KARVO, J., VIRTAMO, J., AALTO, S., AND MARTIKAINEN, O. Blocking of dynamic multicast connections in a single link. In *Proc. of International Broadband Communications Conference, Future of Telecommunications* (Stuttgart, Germany, Apr. 1998), pp. 473–483.

[2] KARVO, J., MARTIKAINEN, O., VIRTAMO, J., AND AALTO, S. Blocking of dynamic multicast connections. *Telecommunication Systems 16*, 3,4 (2001), 467–481.

[3] KARVO, J., AND AALTO, S. Average signalling load for multicast group management. In *Proc. International Teletraffic Congress ITC-16* (Edinburgh, Great Britain, June 1999), D. Smith and P. Key, Eds., pp. 509–518.

[4] AALTO, S., KARVO, J., AND VIRTAMO, J. Calculating blocking probabilities in multicast loss systems. In *Proc. Intl Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2002)* (San Diego, CA, July 2002), pp. 833–842.

[5] KARVO, J., AALTO, S., AND VIRTAMO, J. Blocking probabilities of two-layer statistically indistinguishable multicast streams. In *Proc. International Teletraffic Congress ITC-17* (Salvador da Bahia, Brazil, Sept. 2001), J. M. de Souza, N. L. S. Fonseca, and E. A. de Souza e Silva, Eds., pp. 769–779.

[6] KARVO, J., AALTO, S., AND VIRTAMO, J. Blocking probabilities of multi-layer multicast streams. In *2002 Workshop on High Performance Switching and Routing (HPSR 2002)* (Kobe, Japan, May 2002), pp. 268–277.

[7] LASSILA, P., KARVO, J., AND VIRTAMO, J. Efficient importance sampling for Monte Carlo simulation of multicast networks. In *Proc. INFOCOM'01* (Anchorage, Alaska, Apr. 2001), pp. 432–439.

[8] KARVO, J. Efficient simulation of blocking probabilities for multi-layer multicast streams. In *Proc. IFIP Networking 2002* (Pisa, Italy, May 2002), E. Gregori, M. Conti, A. T. Campbell, G. Omidyar, and M. Zukerman, Eds., vol. 2345 of *Lecture Notes in Computer Science (LNCS)*, pp. 1020–1031.

[9] AALTONEN, J., KARVO, J., AND AALTO, S. Multicasting vs. unicasting in mobile communication systems. In *Proc. Workshop on Wireless Mobile Multimedia, WoWMoM 2002* (Atlanta, GA, Sept. 2002), pp. 104–108.

[10] AALTO, S., AND VIRTAMO, J. Combinatorial algorithm for calculating blocking probabilities in multicast networks. In *Proc. 15th Nordic Teletraffic Seminar, NTS-15* (Lund, Sweden, Aug. 2000), pp. 23–34.

[11] Adamic, L. A., and Huberman, B. A. The nature of markets in the World Wide Web. *Quarterly Journal of Electronic Commerce 1*, 1 (2000), 5–12.

[12] Aein, J. M. A multi-user-class, blocked-calls-cleared, demand access model. *IEEE Transactions on Communications COM-26*, 3 (Mar. 1978), 378–385.

[13] Armitage, G. *Support for Multicast over UNI 3.0/3.1 based ATM Networks*, Nov. 1996. RFC 2022.

[14] Armitage, G. *VENUS - Very Extensive Non-Unicast Service*, Sept. 1997. RFC 2191.

[15] Armitage, G. *Using the MARS Model in non-ATM NBMA Networks*, Jan. 1998. RFC 2269.

[16] Armstrong, S., Freier, A., and Marzullo, K. *Multicast Transport Protocol*, Feb. 1992. RFC 1301.

[17] Bafutto, M., Kühn, P. J., and Willmann, G. Capacity and performance analysis of signaling networks in multivendor environments. *IEEE Journal on Selected Areas in Communications 12*, 3 (Apr. 1994), 490–500.

[18] Bosch i Pérez, E. Performance evaluation of multicast systems. Master's thesis, Technical University of Denmark, Apr. 1998.

[19] Boucherie, R. J., and van Dijk, N. M. On the arrival theorem for product form queueing networks with blocking. *Performance Evaluation 29*, 3 (1997), 155–176.

[20] Boussetta, K., and Belyot, A.-L. Multirate resource sharing for unicast and multicast connections. In *Proceedings of Broadband Communications'99* (Hong Kong, Nov. 1999), D. H. K. Tsang and P. J. Kühn, Eds., pp. 561–570.

[21] Breslau, L., Cao, P., Fan, L., Phillips, G., and Shenker, S. Web caching and Zipf-like distributions: Evidence and implications. In *Proc. INFOCOM'99* (New York, USA, Mar. 1999), vol. 1, pp. 126–134.

[22] Chan, W. C., and Geraniotis, E. Tradeoff between blocking and dropping in multicasting networks. In *ICC '96 Conference Record* (June 1996), vol. 2, pp. 1030–1034.

[23] Chao, X., and Miyazawa, M. On truncation properties of finite-buffer queues and queueing networks. *Probability in the Engineering and Informational Sciences 14* (2000), 409–423.

[24] Chlebus, E., Zbiezek, T., and Ludwin, W. Analysis of channel holding time in wireless mobile systems: Does the probability distribution of cell residence time matter? In *Proc. International Teletraffic Congress, ITC-16* (1999), P. Key and D. Smith, Eds., pp. 117–128.

[25] CHOUDHURY, G. L., LEUNG, K. K., AND WHITT, W. An algorithm to compute blocking probabilities in multi-rate multi-class multi-resource loss models. *Adv. Appl. Prob. 27* (Dec. 1995), 1104–1143.

[26] CHOUDHURY, G. L., LEUNG, K. K., AND WHITT, W. An inversion algorithm to complete blocking probabilities in loss networks with state-dependent rates. *IEEE/ACM Transactions on Networking 3*, 5 (Oct. 1995), 585–601.

[27] CHUNG, S.-P., AND ROSS, K. W. Reduced load approximations for multirate loss networks. *IEEE Transactions on Communications 41*, 8 (Aug. 1993), 1222–1231.

[28] COHEN, J. W. The generalized Engset formulae. *Philips Telecommunication Review 18*, 4 (Nov. 1957), 158–170.

[29] DARTOIS, J. P. Lost call cleared systems with unbalanced traffic sources. In *Proc. International Teletraffic Congress ITC-6* (München, Germany, 1970), pp. 215/1–7.

[30] DELBROUCK, L. E. N. A unified approximate evaluation of congestion functions for smooth and peaky traffics. *IEEE Transactions on Communications COM-29*, 2 (Feb. 1981), 85–91.

[31] DELBROUCK, L. E. N. On the steady-state distribution in a service facility carrying mixtures of traffic with different peakedness factors and capacity requirements. *IEEE Transactions on Communications COM-31*, 11 (Nov. 1983), 1209–1211.

[32] DIOT, C., DABBOUS, W., AND CROWCROFT, J. Multipoint communication: A survey of protocols, functions, and mechanisms. *IEEE Journal on Selected Areas in Communications 15*, 3 (Apr. 1997), 277–290.

[33] DZIONG, Z. A method for calculation of traffic carried on network paths. In *Proc. International Teletraffic Congress, ITC-10* (Montréal, Canada, 1983), vol. 2, pp. 4.3b – 7NR.

[34] ENGSET, T. Die Wahrscheinlichkeitsrechnung zur Bestimmung der Wähleranzahl in automatischen Fernsprechämtern. *Elektrotechnische Zeitschrift 39*, 31 (Aug. 1918), 304–306.

[35] ENGSET, T. On the calculation of switches in an automatic telephone system. *Telektronikk*, 2 (1998). Translation from a 1915 technical report.

[36] ERLANG, A. K. Solution of some problems in the theory of probabilities of significance in automatic telephone exchanges. In *The Life and Works of A. K. Erlang* (Copenhagen, 1960), E. Brockmeyer, H. L. Halstrøm, and A. Jensen, Eds., vol. 287 of *Acta Polytechnica Scandinavica*, pp. 138–155. First published in *Elektroteknikeren* Vol. 13 (1917).

[37] FENNER, W. *Internet Group Management Protocol, Version 2*, Nov. 1997. RFC 2236.

[38] FORTET, R., AND GRANDJEAN, C. Congestion in a loss system when some calls want several devices simultaneously. *Electrical Communication 39*, 4 (1964), 513–526.

[39] GIMPELSON, L. A. Analysis of mixtures of wide- and narrow-band traffic. *IEEE Transactions on Communication Technology 13*, 3 (Sept. 1965), 258–266.

[40] IVERSEN, V. B. The exact evaluation of multi-service loss systems with access control. *Teleteknik, English Edition 31*, 2 (Aug. 1987), 56–61.

[41] JENSEN, A. An elucidation of Erlang's statistical works through the theory of stochastic processes. In *The Life and Works of A. K. Erlang* (Copenhagen, 1960), E. Brockmeyer, H. L. Halstrøm, and A. Jensen, Eds., vol. 287 of *Acta Polytechnica Scandinavica*, pp. 23–100. The first edition published as Transactions of the Danish Academy of Technical Sciences, 1948, Nr. 2.

[42] JORDÁN, J., AND BARCELÓ, F. Statistical modelling of channel occupancy in trunked PAMR systems. In *Proc. International Teletraffic Congress, ITC-15* (Washington, USA, June 1997), V. Ramaswami and P. E. Wirth, Eds., pp. 1169–1178.

[43] KANT, K., AND ONG, L. Signaling in emerging telecommunications and data networks. *Proceedings of the IEEE 85*, 10 (Oct. 1997), 1612–1621.

[44] KARVO, J. Generalized Engset system – properties and an application. In *Workshop on Modelling, Measuring and Quality of Service – Proc. of the 7th Summer School on Telecommunications* (Lappeenranta, Finland, Aug. 1998), J. Jormakka, Ed., pp. 88–97.

[45] KAUFMAN, J. S. Blocking in a shared resource environment. *IEEE Transactions on Communications com-29*, 10 (Oct. 1981), 1474–1481.

[46] KELLY, F. P. *Reversibility and Stochastic Networks*. John Wiley & Sons, 1979.

[47] KLEINROCK, L. *Queueing systems; Volume 1: Theory*. John Wiley & Sons, New York, 1975.

[48] KOLMOGOROV, A. Zur Theorie der Markoffschen Ketten. *Mathematische Annalen 112* (1936), 155–160.

[49] KUCZURA, A., AND BAJAJ, D. A method of moments for the analysis of a switched communication network's performance. *IEEE Transactions on Communications COM-25*, 2 (Feb. 1977).

[50] LASSILA, P. E., AND VIRTAMO, J. T. Efficient importance sampling for Monte Carlo simulation of loss systems. In *Proceedings of the ITC-16* (June 1999), Elsevier, pp. 787–796.

[51] LASSILA, P. E., AND VIRTAMO, J. T. Nearly optimal importance sampling for Monte Carlo simulation of loss systems. *ACM Transactions on Modeling and Computer Simulation 10*, 4 (Oct. 2000), 326–347.

[52] LIN, P. M., LEON, B. J., AND STEWART, C. R. Analysis of circuit-switched networks employing originating-office control with spill-forward. *IEEE Transactions on Communications COM-26*, 6 (June 1978), 754–765.

[53] MANFIELD, D. R., AND DOWNS, T. Decomposition of traffic in loss systems with renewal input. *IEEE Transactions on Communications COM-27*, 1 (Jan. 1979), 44–58.

[54] NAOUMOV, V. Normal-type approximation for multi-service systems with trunk reservation. *Telecommunication Systems*, 4 (1995), 113–118.

[55] NILSSON, A. A., PERRY, M., GERSHT, A., AND IVERSEN, V. B. On multi-rate Erlang-B computations. In *Teletraffic Engineering in a Competitive World, Proc. ITC-16* (Edinburgh, Scotland, June 1999), P. Key and D. Smith, Eds., pp. 1051–1060.

[56] NYBERG, E. Calculation of blocking probabilities and dimensioning of multicast networks. Master's thesis, Networking Laboratory, Helsinki University of Technology, Oct. 1999.

[57] NYBERG, E., VIRTAMO, J., AND AALTO, S. An exact end-to-end blocking probability algorithm for multicast networks. *Submitted for publication*.

[58] NYBERG, E., VIRTAMO, J., AND AALTO, S. An exact algorithm for calculating blocking probabilities in multicast networks. In *Networking 2000* (Paris, France, May 2000), G. Pujolle, H. Perros, S. Fdida, U. Körner, and I. Stavrakakis, Eds., pp. 275–286.

[59] ORLIK, P. V., AND RAPPAPORT, S. S. A model for teletraffic performance and channel holding time characterization in wireless cellular communication with general session and dwell time distributions. *IEEE Journal on Selected Areas in Communications*, 5 (June 1998), 788–803.

[60] PALM, C. Analysis of the Erlang traffic formulæ for busy-signal arrangements. *Ericsson Technics*, 4 (1938), 39–58.

[61] PIÓRO, M., KÖRNER, U., AND WALLSTRÖM, B. Design methods and routing control in integrated services networks with alternative routing. In *Proc. International Teletraffic Congress, ITC-12* (Torino, Italy, 1988), vol. 5, pp. 5.4A.4.1–5.4A.4.7.

[62] PIÓRO, M., LUBACZ, J., AND KÖRNER, U. Traffic engineering problems in multiservice circuit switched networks. *Computer Networks and ISDN Systems 20*, 1–5 (Dec. 1990), 127–136. Special Volume, ITC Specialist Seminar, 25–29 Sep, 1989, Adelaide, Australia.

[63] ROBERTS, J. W. A service system with heterogenous user requirements – application to multi-services telecommunications systems. In *Performance of data communication systems and their applications* (Amsterdam, Sept. 1981), G. Pujolle, Ed., North-Holland, pp. 423–431.

[64] ROSS, K. W. *Multiservice Loss Models for Broadband Telecommunication Networks.* Springer Verlag, London, 1995.

[65] RYKOV, V. V., AND SAMOUYLOV, K. E. Product form solution for stochastic model of dynamic multicast connections. In *Intl. Conference on Distributed Computer Communication Networks: Theory and Applications* (Tel Aviv, Israel, Nov. 1999), pp. 152–160.

[66] SAMOUYLOV, K., AND BOBRIKOV, A. On calculating the probability measures for multicasting networks. In *Proc. International Seminar on Telecommunication Networks and Teletraffic Theory* (St. Petersburg, Russia, Jan. 2002), pp. 104–115.

[67] SAMOUYLOV, K., AND GAIDAMAKA, Y. Analytical model of multicast networks and single link performance analysis. In *Proc. ConTEL 2001* (Zagreb, Croatia, 2001).

[68] SAMUDRA, P., Ed. *ATM User-Network Interface (UNI) Signalling Specification, Version 4.0.* The ATM Forum, July 1996.

[69] SCHASSBERGER, R. Insensitivity of steady-state distributions of generalized semi-Markov processes with speeds. *Adv. Appl. Prob. 10* (1978), 836–851.

[70] SCHASSBERGER, R. Two remarks on insensitive stochastic models. *Adv. Appl. Prob. 18* (1986), 791–814.

[71] TSANG, D. H. K., AND ROSS, K. W. Algorithms to determine exact blocking probabilities for multirate tree networks. *IEEE Transactions on Communications 38*, 8 (Aug. 1990), 1266–1271.

[72] WHITT, W. Continuity of generalized semi-Markov processes. *Mathematics of Operations Research 5*, 4 (Nov. 1980), 494–501.

[73] WHITTLE, P. *Systems in Stochastic Equilibrium.* John Wiley and Sons, Chichester, 1986.

[74] WHITTLE, P. Applied probability in Great Britain. *Operations Research 50*, 1 (Jan.–Feb. 2002), 227–239.

[75] WOLFF, R. W. Poisson arrivals see time averages. *Operations Research 30*, 2 (Mar.–Apr. 1982), 223–231.

Publication [1]

KARVO, J., VIRTAMO, J., AALTO, S., AND MARTIKAINEN, O. Blocking of dynamic multicast connections in a single link. In *Proc. of International Broadband Communications Conference, Future of Telecommunications* (Stuttgart, Germany, Apr. 1998), pp. 473–483.

Publication [2]

KARVO, J., MARTIKAINEN, O., VIRTAMO, J., AND AALTO, S. Blocking of dynamic multicast connections. *Telecommunication Systems 16*, 3,4 (2001), 467–481.

Publication [3]

KARVO, J., AND AALTO, S. Average signalling load for multicast group management. In *Proc. International Teletraffic Congress ITC-16* (Edinburgh, Great Britain, June 1999), D. Smith and P. Key, Eds., pp. 509–518.

Publication [4]

AALTO, S., KARVO, J., AND VIRTAMO, J. Calculating blocking probabilities in multicast loss systems. In *Proc. Intl Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2002)* (San Diego, CA, July 2002). pp. 833–842.

Publication [5]

KARVO, J., AALTO, S., AND VIRTAMO, J. Blocking probabilities of two-layer statistically indistinguishable multicast streams. In *Proc. International Teletraffic Congress ITC-17* (Salvador da Bahia, Brazil, Sept. 2001), J. M. de Souza, N. L. S. Fonseca, and E. A. de Souza e Silva, Eds., pp. 769–779.

Publication [6]

KARVO, J., AALTO, S., AND VIRTAMO, J. Blocking probabilities of multi-layer multicast streams. In *2002 Workshop on High Performance Switching and Routing (HPSR 2002)* (Kobe, Japan, May 2002), pp. 268–277.

Publication [7]

LASSILA, P., KARVO, J., AND VIRTAMO, J. Efficient importance sampling for Monte Carlo simulation of multicast networks. In *Proc. INFOCOM'01* (Anchorage, Alaska, Apr. 2001), pp. 432–439.

Publication [8]

KARVO, J. Efficient simulation of blocking probabilities for multi-layer multicast streams. In *Proc. IFIP Networking 2002* (Pisa, Italy, May 2002), E. Gregori, M. Conti, A. T. Campbell, G. Omidyar, and M. Zukerman, Eds., vol. 2345 of *Lecture Notes in Computer Science (LNCS)*, pp. 1020–1031.

Publication [9]

AALTONEN, J., KARVO, J., AND AALTO, S. Multicasting vs. unicasting in mobile communication systems. In *Proc. Workshop on Wireless Mobile Multimedia, WoWMoM 2002* (Atlanta, GA, Sept. 2002), pp. 104–108.