ELSEVIER

Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Performance Evaluation 54 (2003) 311–330

PERFORMANCE EVALUATION
An International Journal

www.elsevier.com/locate/peva

An exact end-to-end blocking probability algorithm for multicast networks

Eeva Nyberg*, Jorma Virtamo, Samuli Aalto

Networking Laboratory, Helsinki University of Technology, P.O. Box 3000, FIN-02015 HUT, Finland

Received 8 September 2000; received in revised form 3 December 2002

Abstract

We consider the calculation of blocking probabilities in multicast trees with dynamic membership. We extend the work by Karvo et al., where an approximate algorithm based on the reduced load approximation (RLA) was given to calculate end-to-end blocking for infinite sized user populations in multicast networks. The new algorithm for calculating end-to-end call blocking exactly for an arbitrary sized user population is based on the known blocking probability algorithm in hierarchical multiservice access networks, where link occupancy distributions are alternately convolved and truncated. We show that the algorithm can be applied to multicast trees embedded in a network with an arbitrary topology carrying also non-multicast traffic. The resource sharing of multicast connections, however, requires the modification of the algorithm by introducing a new type of convolution, the OR-convolution. In addition, we discuss several different user population models for which the algorithm is applicable.

Keywords: Multicast; Blocking; Network; OR-convolution; One-to-many; Dynamic membership

1. Introduction

A multicast transmission originates at a source and propagates along a multicast tree to the set of receivers, hereafter referred to as users. The membership in the multicast tree is assumed dynamic, i.e. a user may leave the tree and new users may request to be joined to the tree at any time. In contrast to a unicast transmission, at the network nodes a single copy of the information stream is delivered to each branch leading to at least one user. The transmission reaches many different end-users without replication of the same information stream separately for each user, thus resulting in bandwidth saving. This kind of transmission is particularly suited to distribution type applications, such as distribution of radio or TV programs, or, e.g., push type services in 3G mobile networks, where certain information is delivered to all the subscribers of the service. The multicast tree can be embedded in a larger network, where also other type of traffic is carried. For instance, one can think of the multicast tree being embedded in an

* Corresponding author.
E-mail addresses: eeva.nyberg@hut.fi (E. Nyberg), jorma.virtamo@hut.fi (J. Virtamo), samuli.aalto@hut.fi (S. Aalto).

0166-5316/$ – see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0166-5316(03)00093-2

ATM network carrying also unicast calls, or in the Internet carrying streaming type applications for which capacity reservations are made using, e.g., the RSVP protocol.

Blocking occurs in a network when, due to limited capacity, at least one link on the route is not able to admit a new call. Traditional mathematical models to calculate blocking probabilities in tree-structured networks exist for unicast traffic. Due to different resource usage, these models cannot directly be used for multicast networks. Only recently, have mathematical models to calculate blocking probabilities in multicast networks been studied. As usual, one has to make a distinction between the time blocking and the call blocking probabilities. The former refers to the fraction of time the system spends in such states where a given type of call could not be admitted, whereas the call blocking probability refers to the fraction of actual call requests that are rejected. If calls arrive according to a Poisson process, these two quantities are equal due to the PASTA property of the Poisson process. In finite user population models, such as studied in this paper, this is not the case, and one must carefully distinguish between these quantities.

The past research has mainly been focused on blocking probabilities in multicast capable switches. Kim [6] studied blocking probabilities in a multirate multicast switch. Three stage switches were studied by Yang and Wang [13] and Listanti and Veltri [7]. Stasiak and Zwierzykowski [12] studied blocking in an ATM node with multicast switching nodes carrying different multirate traffic (unicast and multicast), using Kaufman–Roberts recursion and reduced load approximation (RLA). Admission control algorithms were studied in [10].

Chan and Geraniotis [2] have studied blocking due to finite capacity in network links. They formulated a closed form expression for time blocking probabilities in a network transmitting layered video signals. The model is a multipoint-to-multipoint model. The network consists of several video sources, where each source node can also act as a receiver. The video signals are coded into different layers defining the quality of the video signal received by the user. The traffic class is defined by the triplet: physical path (p), source node (s), and class of video quality (t). The behavior of each user is modeled as a two-state Markov chain, with unique transition rates defined for each traffic class triplet.

Karvo et al. [3,4] studied blocking in a point-to-multipoint network with only one source node. The source is called the service center and it can offer a variety of channels, e.g., TV-channels. The users subscribing to the network may, at any time, join or leave any of the several multicast trees, each carrying a separate multicast transmission or channel offered by the source. The behavior of the user population defines the state probabilities at the links of the tree-structured network. The user population is assumed infinite and the requests to join the network arrive as from a Poisson process. The model studied in [3] considered a simplified case where all but one link in a network have infinite capacity. An exact algorithm was derived to calculate the call blocking probability in this simplified case. Extending the model to the whole network was done only approximately in [4], where end-to-end blocking probabilities were estimated using the RLA approach. The single link case was further broadened by Bousseta and Beylot [1] by including both multirate multicast and unicast traffic in their formulation.

In [8] the single link case discussed in [3,4] was extended to a mathematical model for a multicast network with any number of finite capacity links and an infinite user population. Furthermore, the case of having background traffic on the links of the network was also discussed, independently of [1]. The aim of the present paper is to collect the pieces together and give a unified and more detailed account of the algorithm. Additionally, we extend the algorithm by allowing arbitrary sized, i.e. also finite, user populations and by introducing different user models. A proof for the insensitivity properties of the results is also provided. The new material is presented in Sections 4 and 6 and in Appendix A. Section 7 includes a more detailed justification of the embedded network truncation operators.

This paper continues with Section 2 where the notation used throughout the paper is presented. We also define the leaf link state and state space, and show how the network state can be obtained from the leaf link states. In Section 3 we assume user independence. This is a natural assumption for distribution type applications, where the users do not interact with each other. We then show how the link distributions within the network can be obtained from the leaf link distributions via a new convolution operation, the OR-convolution. In Section 4, four different user population models are introduced and the resulting leaf link distributions are given.

We start the derivation of the algorithm, by separating the tree-structured multicast transmissions from the surrounding distribution network. The first main result is presented in Section 5. It gives an expression for the time blocking probability in a network with any number of finite capacity links, and an algorithm for calculating this blocking probability exactly is introduced. The section also presents some issues related to computational effort. In Section 6 it is shown how the algorithm can be applied for calculating call blocking probabilities using different user models. With the algorithm derived in the simplified setting, the algorithm is easily extended to include non-multicast traffic originating from outside the tree-structured transmission network. The final result, the extension of the algorithm to calculate blocking probabilities in multicast trees embedded in a network with an arbitrary topology carrying also non-multicast traffic is presented in Section 7. Section 8 summarizes the main results and discusses the topics for further research. A proof of the insensitivity properties of the results is given in Appendix A.

2. Network model

In Sections 2–6, we consider the tree-structured subnetwork formed by the routed multicast connections originating from the source. In Section 7, we consider embedding the dynamic multicast tree network in an arbitrary structured network. Until then, we use the term network to refer to the tree-structured multicast network.

2.1. Notation

The notation used throughout this paper is as follows. The set of all links j is denoted by $\mathcal{J} = \{1, \dots, J\}$. Link J refers to the link connecting the source to the rest of the network. Furthermore, let $\mathcal{U} = \{1, \dots, U\} \subset \mathcal{J}$ denote the set of leaf links. The leaf link and user population connected through the leaf link is indexed by $u \in \mathcal{U}$. The set of links on the route from leaf link u to the source is denoted by $\mathcal{R}_u$. $\mathcal{M}_j$ and $\mathcal{N}_j$ stand for the set of all links downstream from link j including link j and the set of downstream links connected to link j, respectively. The set of user populations downstream from link j is denoted by $\mathcal{U}_j$. Note that $\mathcal{U}_j$ is also the set of leaf links downstream from link j, including link j if link $j \in \mathcal{U}$, in other words $\mathcal{U}_j = \mathcal{M}_j \cap \mathcal{U}$. The set of channels i offered by the source is denoted by $\mathcal{I} = \{1, \dots, I\}$. Let $\mathbf{d} = \{d_i; i \in \mathcal{I}\}$, where d_i is the capacity requirement of channel i. Here we assume that the capacity requirements depend only on the channel, but link dependencies could also be included into the model. The capacity of the link j is denoted by C_j. The different sets are depicted in Fig. 1.

Note that we have specified neither the size nor the traffic process of the user population. We will postpone this discussion until Section 4 and start by defining the network state, state space and steady-state probabilities in terms of an arbitrary leaf link process.

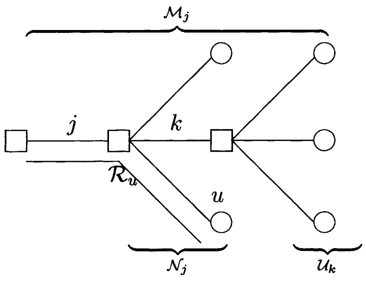

Fig. 1. Example routed multicast connections to show the notation used.

2.2. Link and network state

Let the pair $(u, i) \in \mathcal{U} \times \mathcal{I}$ denote a traffic class, also called a connection. The connection state, which may be off (0) or on (1), is denoted by $Y_{u,i} \in \{0, 1\}$. The state vector $\mathbf{Y}_u = (Y_{u,i}; i \in \mathcal{I}) \in \mathcal{S}$ defines the joint state of different channels on leaf link $u \in \mathcal{U}$, where $\mathcal{S} = \{0, 1\}^I$ denotes the link state space. Similarly, for any link $j \in \mathcal{J}$ the link state is denoted by the vector $\mathbf{Y}_j = (Y_{j,i}; i \in \mathcal{I}) \in \mathcal{S}$.

Consider now a network with all links having infinite capacity. The leaf link states $\mathbf{Y}_u$ jointly define the network state $\mathbf{X}$,

$$\mathbf{X} = (\mathbf{Y}_u; u \in \mathcal{U}) = (Y_{u,i}; u \in \mathcal{U}, i \in \mathcal{I}) \in \Omega, \tag{1}$$

where $\Omega = \{0, 1\}^{U \times I}$ denotes the network state space.

2.2.1. OR-operation

In a tree-structured multicast network, where traffic has resource sharing characteristics, the link states are obtained from the leaf link states using the OR-operation. Consider only two links $s, t \in \mathcal{N}_v$ immediately downstream from link v, where $s, t, v \in \mathcal{J}$. Let $\mathbf{y}_s, \mathbf{y}_t, \mathbf{y}_v \in \mathcal{S}$ denote the states of these three links, respectively. Channel i is idle in link v if it is idle in both links s and t and active in all other cases, which is equivalent to the binary OR-operation. In other words, for $\mathbf{y}_s, \mathbf{y}_t \in \mathcal{S}$

$$\mathbf{y}_v = \mathbf{y}_s \oplus \mathbf{y}_t \in \mathcal{S}, \tag{2}$$

where the vector operator $\oplus$ denotes the OR-operation taken componentwise.

In a multicast link, the link state depends on the user states downstream from the link. If a channel is idle in all links downstream from link j it is off in link j and in all other cases the channel is active. The OR-operation gives the link state $\mathbf{Y}_j = (Y_{j,i}; i \in \mathcal{I}) \in \mathcal{S}$, $j \in \mathcal{J}$ as a function of the network state, $\mathbf{X}$,

$$\mathbf{Y}_j = \mathbf{g}_j(\mathbf{X}) \equiv \bigoplus_{k \in \mathcal{U}_j} \mathbf{Y}_k = \begin{cases} \mathbf{Y}_j & \text{if } j \in \mathcal{U}, \\ \bigoplus\limits_{k \in \mathcal{N}_j} \mathbf{Y}_k & \text{otherwise.} \end{cases}$$

Here, the last form is given to motivate the derivation of the recursive algorithm presented in Section 5.1. Note that, when $\mathbf{X} = \mathbf{x}$ the occupied capacity on the link j is $\mathbf{d} \cdot \mathbf{g}_j(\mathbf{x})$.

When the capacities of one or more links in the network are finite, the network state space Ω is truncated according to the capacity restrictions on each link $j \in \mathcal{J}$. The truncated state space, denoted by $\tilde{\Omega}$, is defined as follows:

$$\tilde{\Omega} = \{\mathbf{x} \in \Omega | \mathbf{d} \cdot \mathbf{g}_j(\mathbf{x}) \le C_j, \forall j \in \mathcal{J}\}.$$

Correspondingly, we denote by $\tilde{\mathbf{X}} \in \tilde{\Omega}$ the state vector in the truncated space.

3. Steady-state distribution in a network with infinite link capacities

The network state is jointly defined by the leaf link states. Under the assumption that each user population $u \in \mathcal{U}$ is independent, and that the link capacities are infinite for all links in the network, the stationary distribution of the network can be obtained from the leaf link distributions, defined by the user population connected through the leaf link. Let us assume that the leaf link distributions, $\pi_u(\mathbf{y}_u) = P(\mathbf{Y}_u = \mathbf{y}_u)$, $u \in \mathcal{U}$, are known. For the whole network, the state probability has a product form,

$$\pi(\mathbf{x}) = P(\mathbf{X} = \mathbf{x}) = \prod_{u \in \mathcal{U}} \pi_u(\mathbf{y}_u), \tag{3}$$

as the user populations are independent.

3.1. OR-convolution

In Section 2.2, it was shown that the link state is obtained by an OR-operation over all downstream leaf link states. Under the assumption of independent user populations and infinite link capacities, the link distributions can be obtained using a new convolution operation, the OR-convolution.

The OR-convolution, denoted by $\otimes$, is the operation,

$$[f_s \otimes f_t](\mathbf{y}_v) = \sum_{\mathbf{y}_s \oplus \mathbf{y}_t = \mathbf{y}_v} f_s(\mathbf{y}_s) f_t(\mathbf{y}_t)$$

defined for any real valued functions f_s and f_t.

The link state distribution is obtained by OR-convolving the appropriate leaf link distributions. Thus, the link state probability, denoted by $\pi_j(\mathbf{y})$, for $\mathbf{y} \in \mathcal{S}$, is equal to

$$\pi_j(\mathbf{y}) = P(\mathbf{Y}_j = \mathbf{y}) = \left[\bigotimes_{k \in \mathcal{U}_j} \pi_k\right](\mathbf{y}) = \begin{cases} \pi_j(\mathbf{y}) & \text{if } j \in \mathcal{U}, \\ \left[\bigotimes_{k \in \mathcal{N}_j} \pi_k\right](\mathbf{y}) & \text{otherwise.} \end{cases}$$

4. User population models

In the previous section, the steady-state distribution of the network was defined in terms of the leaf link distributions. Recall that, according to our notation, the leaf link and the user population connected through the leaf link are equivalent. Consequently, the user population model defines the leaf link distribution $\pi_u(\mathbf{y})$. For the derivation of the blocking algorithm to follow, we further need to assume that the behavior of the user population is described by a reversible Markov process, i.e. a Markov process satisfying the detailed balance equations [5]. We are able to loosen the assumption by allowing general holding time distributions leading to more general processes, see Appendix A for the proof of this insensitivity property.

In this section, we present four different user population models. We first consider a model for a single user choosing from the set of channels $\mathcal{I}$. The second model, presented in Section 4.2 is constructed as a special case of the single user model, the single user being only connection-specific, i.e. having only the possibility of choosing a given channel i. We construct the leaf link distribution by combining the I single users, one for each channel. The most general user population model, the finite user population model, is presented in Section 4.3. It is a model for a population consisting of N users each having the whole selection of channels to choose from. We show how the steady-state probability for the population and thus for the leaf link can be obtained with the aid of the single user model presented in Section 4.1. In Section 4.4, we show how the user population for a finite number of users results in the infinite user population, presented in [3], as the number of users N tends to infinity. The given four user population models and the corresponding four leaf link distributions cover a large variety of user models for different distribution type multicast services, e.g., distribution of radio or TV channels. Furthermore, each model can be defined in terms of the single user model presented next.

4.1. *The single user*

First, we consider a model where each user population consists of a single user. Let $u \in \mathcal{U}$. User u, connected through leaf link u, can either be in the idle state 0 or connected to some channel $i \in \mathcal{I}$. The model proposed here is a Markov process with $I+1$ states. All transitions by user u are made via the idle state. The transition rate from state 0 to state $i \in \mathcal{I}$ is denoted by $\lambda_{u,i} = \alpha_i \lambda_u$, where α_i is the probability of choosing channel i among the channel set $\mathcal{I}$, $\sum_{i\in\mathcal{I}} \alpha_i = 1$. The transition rate from state i to state 0 is denoted by μ_i. The state transition diagram of the Markov process is shown in Fig. 2.

The steady-state probabilities of this single user system are

$$\pi_{u,i} = \rho_{u,i}\pi_{u,0}, \qquad \pi_{u,0} = \left[1 + \sum_{i=1}^{I} \rho_{u,i}\right]^{-1},$$

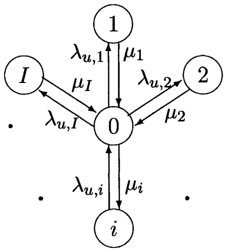

Fig. 2. The Markov process used to model user behavior.

where $\rho_{u,i} = \alpha_i \lambda_u / \mu_i$. Because the state diagram of the model was chosen to be a tree, the detailed balance equations are satisfied, which can also be seen directly

$$\pi_{u,0}\lambda_{u,i} = \pi_{u,i}\mu_i, \quad i \in \mathcal{I}.$$

Thus the process is reversible. Furthermore, it can be shown (cf. Appendix A) that the insensitivity property applies and the channel holding times as well as the user idle times can be generally distributed with means $1/\mu_i$ and $1/\lambda_u$, respectively, leading to a semi-Markov process.

The probability P_u that user u connects to some channel in the multicast network is

$$P_u = 1 - \pi_{u,0} = \frac{\sum_{k\in\mathcal{I}} \rho_{u,k}}{1 + \sum_{k\in\mathcal{I}} \rho_{u,k}}. \tag{4}$$

In addition, the parameter $\hat{\alpha}_i$ is defined as the conditional probability of being in state i given that the user connects to the multicast network,

$$\hat{\alpha}_i = \frac{\rho_{u,i}}{\sum_{k\in\mathcal{I}} \rho_{u,k}} = \frac{\alpha_i/\mu_i}{\sum_{k\in\mathcal{I}} \alpha_k/\mu_k}, \quad i \in \mathcal{I}. \tag{5}$$

It follows that $\pi_{u,i}$ in terms of P_u and $\hat{\alpha}_i$ is

$$\pi_{u,i} = P_u \hat{\alpha}_i, \quad i \in \mathcal{I},$$

The steady-state probabilities $\pi_u(\mathbf{y})$ for leaf link u then have the following form:

$$\pi_u(\mathbf{y}) = \begin{cases} P_u \hat{\alpha}_i & \text{if } \mathbf{y} = \mathbf{e}_i, i \in I, \\ 1 - P_u & \text{if } \mathbf{y} = \mathbf{0}, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Note that the mean idle time $1/\lambda_u$ can be arbitrarily small, in which case the user switches almost directly from one channel to another. However, by having the idle state, we emphasize the fact that the capacity reservation related to the current channel has to be first released, before a new reservation can be made. Namely, it may happen that the new reservation, if it is larger than the previous one, is not accepted because of the capacity constraints. In our model, the user naturally remains idle in such a case.

4.2. The connection-specific single user

In the paper by Chan and Geraniotis [2], each user of a traffic class (p, s, t) was modeled as a two-state Markov chain with unique transition rates. The user model can be obtained as a special case from the single user model presented in the previous section. Instead of being leaf link specific, i.e. having a selection of channels to choose from, each user is now connection-specific. The user is denoted by the pair (u, i), formerly denoting a connection. Behind each leaf link u there are I users, one for each channel i. The Markov process for the connection-specific user (u, i) are obtained by setting the transition rates $\lambda_{u,j} = \mu_j = 0$, for $j \neq i, i \in \mathcal{I}$. The Markov process is depicted in Fig. 3.

As the channels are independent, the unrestricted steady-state distribution for leaf link u is then

$$\pi_u(\mathbf{y}) = \prod_{i\in\mathcal{I}} p_{u,i}^{y_i}(1 - p_{u,i})^{1-y_i}, \tag{7}$$

where $p_{u,i} = \lambda_{u,i}/(\lambda_{u,i} + \mu_i)$.

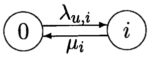

Fig. 3. The Markov process for the connection-specific user.

As the connection-specific user is a special case of the single user, the insensitivity property applies (cf. Appendix A) and the channel holding times as well as the user idle times can be generally distributed with means $1/\mu_i$ and $1/\lambda_{u,i}$, respectively.

4.3. The finite user population

Consider a leaf link u with a user population of size N connected through the link. Users are assumed independent and homogeneous, each user being modeled according to the single user model of Section 4.1. We can obtain the unrestricted leaf link distribution $\pi_u(\mathbf{y})$ in terms of the single user distribution given in (6). To do this, we construct the finite user population of size N from single users, by envisaging that downstream from leaf link u there are N links with infinite capacity each connecting a single user to the network. In other words, we create a new hypothetical set $\mathcal{U}_u$ of users downstream from link u. If each user has the same probability P_u to subscribe to the network and a steady-state distribution given by Eq. (6), then the OR-convolution gives the distribution for the actual leaf link u servicing the population of N users. Thus, the state probability, denoted by $\pi_u(\mathbf{y})$, for $\mathbf{y} \in \mathcal{S}$, is equal to

$$\pi_u(\mathbf{y}) = P(\mathbf{Y}_u = \mathbf{y}) = \left[\bigotimes_{k \in \mathcal{U}_u} \pi_k \right](\mathbf{y}). \tag{8}$$

As the finite user population can be obtained from the single user model, the insensitivity property applies (cf. Appendix A) and the channel holding times as well as the user idle times can be generally distributed with means $1/\mu_i$ and $1/\lambda_{u,i}$, respectively.

The leaf link distribution given in Eq. (8) can also be obtained by calculating the state probabilities of $|\mathcal{U}_u| = N$ users, using a multinomial distribution with parameters $p_i = P_u\hat{\alpha}_i$, for $i \in \mathcal{I}$ and $p_0 = 1 - P_u$,

$$P(\boldsymbol{\Xi} = \xi | \Xi_0 + \cdots + \Xi_I = N) = \begin{cases} N! \prod_{i=0}^{I} \dfrac{p_i^{\xi_i}}{\xi_i!} & \text{if } \xi_0 + \cdots + \xi_I = N, \\ 0 & \text{otherwise}, \end{cases} \tag{9}$$

where $\boldsymbol{\Xi} = (\Xi_i, i = 0, \ldots, I)$ is the state vector, $\Xi_i \in \mathbb{N}$. The state probabilities given in Eq. (8) are obtained by summing the state probabilities of Eq. (9) to take into account the multicast conditions.

4.4. The infinite user population

As the number of users N belonging to user population u tends to infinity, the population model converges to the infinite population model presented by Karvo et al. [3]. This is easily seen, as the multinomial distribution with parameters $p_i = P_u\hat{\alpha}_i$ and expected values $NP_u\hat{\alpha}_i$ converges to the joint distribution of independent Poisson distributed variables with parameter $a_{u,i} = (\hat{\lambda}_u/\mu_i)\alpha_i$. Writing the expected value with the help of Eqs. (4) and (5) gives

$$NP_u\hat{\alpha}_i = N\frac{\lambda_u \sum_{k\in\mathcal{I}}\alpha_k/\mu_k}{1+\lambda_u\sum_{k\in\mathcal{I}}\alpha_k/\mu_k}\frac{\alpha_i/\mu_i}{\sum_{k\in\mathcal{I}}\alpha_k/\mu_k}.$$

The limit of the expected value $NP_u\hat{\alpha}_i$ is then

$$\lim_{N\to\infty}(NP_u\hat{\alpha}_i) = \lim_{N\to\infty}\left(\frac{\alpha_i}{\mu_i}\frac{N\lambda_u}{1+\lambda_u\sum_{k\in\mathcal{I}}\alpha_k/\mu_k}\right) \to \frac{\alpha_i}{\mu_i}\hat{\lambda}_u = a_{u,i}, \quad \forall i \in \mathcal{I},$$

where $\lim_{N\to\infty}N\lambda_u \to \hat{\lambda}_u$.

The finite user population model therefore converges to the infinite user population model presented in [3]. The reversible Markov process for the infinite user population is the joint queue length of I independent $M/M/\infty$ queues. The unrestricted stationary distribution for leaf link u with an infinite sized user population connected through is thus

$$\pi_u(\mathbf{y}) = \prod_{i\in\mathcal{I}}(1-\mathrm{e}^{-a_{u,i}})^{y_i}(\mathrm{e}^{-a_{u,i}})^{1-y_i}. \tag{10}$$

Note further the similarity between Eqs. (7) and (10). In both models the stationary leaf link distribution is the joint distribution of connection-specific user populations. The equations differ only in the probability of connecting to the channel, $\lambda_{u,i}/(\lambda_{u,i}+\mu_i)$ and $1-\mathrm{e}^{-a_{u,i}}$, respectively. For the infinite user population model, the insensitivity property applies (cf. Appendix A) and the channel holding times can be generally distributed with mean $1/\mu_i$, leading to independent $M/G/\infty$ queues.

5. Time blocking in a multicast tree network

When the capacities of one or more links in the network are finite, the network state space is replaced by the truncated network state space $\tilde{\Omega}$. In the previous section we specified four different user models. Assuming independent user populations, i.e. independent leaf link distributions, the state probabilities of the truncated system differ from those of an infinite system only by the normalization constant $G(\tilde{\Omega}) = \sum_{\mathbf{x}\in\tilde{\Omega}}\pi(\mathbf{x})$. This result, known as the truncation principle, applies if the idle and holding time distributions are exponential, as the resulting state vector $\tilde{X}$ is a reversible Markov process (cf. [5]). For general idle and holding time distributions, the applicability of the truncation principle is shown in Appendix A. The state probabilities of the truncated system are therefore

$$\tilde{\pi}(\mathbf{x}) = P(\tilde{\mathbf{X}}=\mathbf{x}) = P(\mathbf{X}=\mathbf{x}|\mathbf{X}\in\tilde{\Omega}) = \frac{\pi(\mathbf{x})}{G(\tilde{\Omega})} \quad \text{for } \mathbf{x}\in\tilde{\Omega}. \tag{11}$$

When the capacities of the links are finite, blocking occurs. A call belonging to traffic class (u, i) is blocked if there is not enough capacity in the network to set up the connection. Note that, once channel i is active on any link belonging to the route R_u of user population u, no extra capacity is required on that link for a new connection (u, i). Let us define another truncated set $\tilde{\Omega}_{u,i} \subset \tilde{\Omega}$ with a tighter capacity restriction for the links on route $\mathcal{R}_u$,

$$\tilde{\Omega}_{u,i} = \{\mathbf{x}\in\Omega|\mathbf{d}\cdot(\mathbf{g}_j(\mathbf{x})\oplus(\mathbf{e}_i 1_{j\in\mathcal{R}_u}))\le C_j, \forall j\in\mathcal{J}\},$$

where $\mathbf{e}_i$ is the I-dimensional vector consisting of only zeros except for a one in the ith component, and $1_{j\in\mathcal{R}_u}$ is the indicator function equal to 1 for $j \in \mathcal{R}_u$ and 0 otherwise. This set defines the states where blocking does not occur when user u requests a connection to channel i. The time blocking probability b_i^{t} for traffic class (u, i) is thus,

$$b_{u,i}^{\mathrm{t}} = 1 - P(\tilde{\mathbf{X}} \in \tilde{\Omega}_{u,i}) = 1 - \frac{G(\tilde{\Omega}_{u,i})}{G(\tilde{\Omega})}. \tag{12}$$

This approach requires calculating two state probability sums: one over the set of non-blocking states appearing in the numerator and another one over the set of allowed states appearing in the denominator of Eq. (12).

The multicast one-to-many connections form a tree-type structure, and much of the theory in calculating blocking probabilities in hierarchical multiservice access networks [9] can be used to formulate the end-to-end blocking probability algorithm in a multicast network as well.

5.1. The algorithm

Using the analogy to tree-structured access networks, the time blocking probability is calculated by recursively convolving the state distributions of individual links proceeding from the leaf links to the origin link, and at each step, truncating the link distributions according to the capacity restriction of the link.

In order to calculate the denominator of Eq. (12), let us define a new subset $\tilde{\mathcal{S}}_j$ of the set of link states, $\mathcal{S}$,

$$\tilde{\mathcal{S}}_j = \{\mathbf{y} \in \mathcal{S} | \mathbf{d} \cdot \mathbf{y} \le C_j\} \quad \text{for } j \in \mathcal{J}.$$

Thus, $\tilde{\mathcal{S}}_j$ refers to those states of link j for which the capacity constraint is satisfied. The corresponding truncation operator acting on any real valued function f is defined as

$$T_j f(\mathbf{y}) = f(\mathbf{y}) 1_{\mathbf{y}\in\tilde{\mathcal{S}}_j}. \tag{13}$$

For $\mathbf{y} \in \mathcal{S}$ let

$$Q_j(\mathbf{y}) = P(\mathbf{Y}_j = \mathbf{y}; \mathbf{Y}_k \in \tilde{\mathcal{S}}_k, \forall k \in \mathcal{M}_j). \tag{14}$$

This is the probability that link j is in state $\mathbf{y}$ and all the links downstream from link j (link j included) satisfy the capacity constraints. It is crucial for our algorithm that these probabilities can be calculated recursively as follows:

$$Q_j(\mathbf{y}) = \begin{cases} T_j \pi_j(\mathbf{y}) & \text{if } j \in \mathcal{U}, \\ T_j \left[\bigotimes_{k\in\mathcal{N}_j} Q_k \right](\mathbf{y}) & \text{otherwise.} \end{cases}$$

Note that, if the capacity constraint of link $j \in \mathcal{M}_j$ is relaxed, then the branches terminating at link j are independent, and the probabilities of the jointly requested channel state can be obtained by the OR-convolution. The effect of the finite capacity C_j of link j is then just the truncation of the distribution to the states for which the requested capacity is no more than C_j.

The state sum $G(\tilde{\Omega})$ needed to calculate the blocking probability in Eq. (12) is equal to

$$G(\tilde{\Omega}) = \sum_{\mathbf{y}\in\mathcal{S}} Q_J(\mathbf{y}),$$

where $Q_J(\mathbf{y})$ is the probability (14) related to link J connecting the source to the rest of the network.

Similarly for the numerator of Eq. (12), let $\tilde{\mathcal{S}}_j^{u,i} \subset \tilde{\mathcal{S}}_j$ be defined as the set of states on link j that do not prevent user u from connecting to multicast channel i, i.e.

$$\tilde{\mathcal{S}}_j^{u,i} = \{\mathbf{y} \in \mathcal{S} | \mathbf{d} \cdot (\mathbf{y} \oplus (\mathbf{e}_i 1_{j\in\mathcal{R}_u})) \le C_j\} \quad \text{for } j \in \mathcal{J}.$$

The truncation operator is then

$$T_j^{u,i} f(\mathbf{y}) = f(\mathbf{y}) 1_{\mathbf{y}\in\tilde{\mathcal{S}}_j^{u,i}}. \tag{15}$$

For $\mathbf{y} \in \mathcal{S}$ let

$$Q_j^{u,i}(\mathbf{y}) = P(\mathbf{Y}_j = \mathbf{y}; \mathbf{Y}_k \in \tilde{\mathcal{S}}_k^{u,i}, \forall k \in \mathcal{M}_j). \tag{16}$$

This is the probability that link j is in state $\mathbf{y}$ and none of the links downstream from link j (link j included) prevents user u from connecting to multicast channel i. It is as crucial as above that also these probabilities can be calculated recursively as follows:

$$Q_j^{u,i}(\mathbf{y}) = \begin{cases} T_j^{u,i} \pi_j(\mathbf{y}) & \text{if } j \in \mathcal{U}, \\ T_j^{u,i} \left[\bigotimes_{k\in\mathcal{N}_j} Q_k^{u,i} \right](\mathbf{y}) & \text{otherwise.} \end{cases}$$

The state sum in the numerator of Eq. (12) is then

$$G(\tilde{\Omega}_{u,i}) = \sum_{\mathbf{y}\in\mathcal{S}} Q_J^{u,i}(\mathbf{y}),$$

where $Q_J^{u,i}(\mathbf{y})$ is the probability (16) related to link J connecting the source to the rest of the network.

Finally, the blocking probability in Eq. (12) is

$$b_{u,i}^{\mathrm{t}} = 1 - \frac{\sum_{\mathbf{y}\in\mathcal{S}} Q_J^{u,i}(\mathbf{y})}{\sum_{\mathbf{y}\in\mathcal{S}} Q_J(\mathbf{y})}.$$

5.2. Computational complexity

The complexity of the algorithm increases exponentially with the number of channels, as the number of states in each of the distributions to be convolved is 2^I. This can be seen by investigating the computational effort of the OR-convolution algorithm. The computational effort of convolving two state vectors of length 2^I is $O((2^I - 1)^2)$. However, the total computational effort of the OR-convolution algorithm in a network with U user populations, $O((U-1)(2^I-1)^2) = O(U2^{2I})$, grows on linearly with respect to U, irrespective of the number of links J. This can be compared to a brute force approach of going through all the 2^{UI}

network states and summing the probabilities of those which satisfy the conditions of an allowed state. Clearly, the computational effort of the OR-convolution algorithm grows exponentially as the number I of channels grows, but it does not depend critically on the size of the network, defined by the number of user populations, as is the case with the brute force method.

6. Call blocking in a multicast tree network

Until now, we have shown how the exact algorithm can be used to calculate time blocking probabilities in point-to-multipoint multicast networks, with arbitrary sized user populations. However, call blocking probability is often of more interest. In [3] call blocking b_i^{c} was defined for the multicast network; call blocking occurs when a user is not able to subscribe to channel i.

6.1. Call blocking with infinite user populations

Due to Poisson arrivals, the time blocking probability for the infinite user population is equal to the call blocking probability. Therefore, no modifications of the algorithm are needed. Note that the single link model by Karvo et al. [3] is a special case of the network model derived in this paper, and hence the same results can be obtained using the network algorithm.

6.2. Call blocking with single users

The Markov process model that describes the behavior of a single user was presented in Section 4.1. The model assumes that each user is subscribed to one channel at a time and a request for a new channel can occur only through the idle state. A user experiences call blocking, when there is not enough capacity to turn the channel on. Call blocking for user u is equal to time blocking in a network where user u is removed. This follows easily from the product form state distribution. Removing user u from the network is equivalent to setting user u in state 0,

$$\pi_u(\mathbf{y}) = \begin{cases} 1 & \text{if } \mathbf{y} = \mathbf{0}, \\ 0 & \text{otherwise}. \end{cases} \tag{17}$$

For all other $j \in \mathcal{U}$ the state probabilities are given by Eq. (6).

As described in Section 3 the leaf link distributions define the network distribution and by using the state probabilities defined above, the algorithms presented in Section 5.1 can be used to calculate the time blocking in the reduced system. The resulting end-to-end call blocking probability is

$$B_{u,i}^{\mathrm{c}} = 1 - \frac{\sum_{\mathbf{y}\in\mathcal{S}} Q_J^{u,i}(\mathbf{y})}{\sum_{\mathbf{y}\in\mathcal{S}} Q_J(\mathbf{y})},$$

where

$$Q_j(\mathbf{y}) = \begin{cases} 1_{\mathbf{y}=\mathbf{0}} & \text{if } j = u,\ j \in \mathcal{U}, \\ T_j \pi_j(\mathbf{y}) & \text{if } j \neq u,\ j \in \mathcal{U}, \\ T_j \left[\bigotimes_{k\in\mathcal{N}_j} Q_k \right](\mathbf{y}) & \text{otherwise}, \end{cases}$$

and

$$Q_j^{u,i}(\mathbf{y}) = \begin{cases} 1_{\mathbf{y}=\mathbf{0}} & \text{if } j = u,\ j \in \mathcal{U}, \\ T_j^{u,i}\pi_j(\mathbf{y}) & \text{if } j \neq u,\ j \in \mathcal{U}, \\ T_j^{u,i}\left[\bigotimes_{k\in\mathcal{N}_j} Q_k^{u,i}\right](\mathbf{y}) & \text{otherwise.} \end{cases}$$

6.3. Call blocking with connection-specific users

The connection-specific user model presented in Section 4.2 is a special case of the single user model, and the call blocking probability is calculated in a similar fashion. The end-to-end call blocking probability for user (u, i) is obtained using the time blocking algorithm to a network where user (u, i) is removed. In other words, by replacing $\pi_u(\mathbf{y})$ with the state probability

$$\pi_u^{(i)}(\mathbf{y}) = \prod_{j\in\mathcal{I}\setminus\{i\}} p_{u,j}^{y_j}(1 - p_{u,j})^{(1-y_j)}.$$

For all the other users, Eq. (7) is used.

6.4. Call blocking with finite user populations

For a finite user population with $N > 1$, the call blocking probability can be calculated by envisaging the underlying single user processes downstream from the leaf link, as was done in Section 4.3, i.e. by imagining that each user is connected to the leaf link by a separate infinite capacity link. The call blocking probability is then equal to the call blocking probability of a single user in this extended system.

7. Blocking in a multicast tree embedded in a larger network

Until now, only the tree-structured part of the network, resulting from the multicast traffic offered by the source was considered. As mentioned earlier, the algorithm can be extended to generally structured networks, with mixed traffic. In this case, the tree-structured distribution portion of the network carries, in addition to multicast traffic, background unicast traffic originating from the surrounding network as illustrated in Fig. 4.

We assume that the background traffic is independent on each link. This is a reasonable assumption in a network carrying a large number of traffic streams, none of which is dominating the others. The distribution of the background traffic is also assumed to be independent of the multicast traffic in the link. This is a viable assumption when the background traffic consists of calls, like calls in an ATM network, or, in the case of the Internet, streams with capacity reservations. The present model does not directly apply to the case where the background traffic is elastic responding to the available capacity.

The non-multicast traffic in link j is assumed to be Poisson with traffic intensity A_j. The capacity requirement is equal to one unit of capacity. The link occupancy distribution of the non-multicast traffic

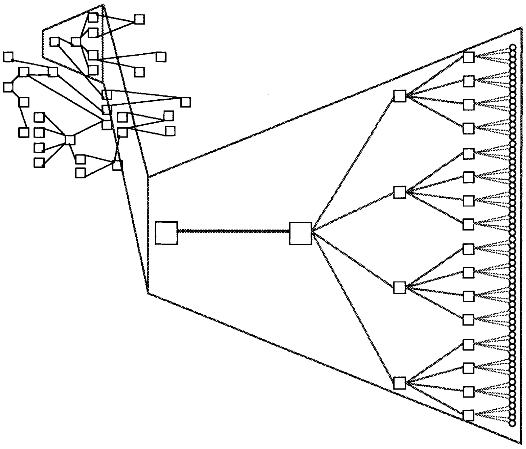

Fig. 4. A distribution network embedded in a general network.

in a link with infinite capacity is thus,

$$q_j(z) = \frac{(A_j)^z}{z!}\,\mathrm{e}^{-A_j}, \quad z \in \mathbb{N}. \tag{18}$$

Here we deal only with single rate unicast traffic, the generalization to multirate Poisson traffic is straightforward and is considered, for example, in [1]. Note also that, for the truncation principle to apply, the background traffic can be modeled by any reversible Markov process or, even, the corresponding non-Markovian process with general holding times. In this work we consider the Poisson case.

The inclusion of background traffic affects only the truncation step of the algorithm presented in Section 5.1. The state probabilities are defined as in Section 4. The state probabilities of the link states that require more capacity than available on the link are set to zero as before. However, also the state probabilities of the states that satisfy the capacity restriction of the link are altered, as the available capacity on the link depends on the amount of non-multicast traffic on the link.

Therefore, the truncation functions presented in Eqs. (13) and (15) must be replaced by the operators

$$\hat{T}_j f(\mathbf{y}) = P(Z_j \le C_j - \mathbf{d}\cdot\mathbf{y}) f(\mathbf{y}) = \sum_{z=0}^{C_j-\mathbf{d}\cdot\mathbf{y}} q_j(z) f(\mathbf{y}),$$

$$\hat{T}_j^{u,i} f(\mathbf{y}) = P(Z_j \le C_j - \mathbf{d}\cdot(\mathbf{y}\oplus(\mathbf{e}_i 1_{j\in\mathcal{R}_u}))) f(\mathbf{y}) = \sum_{z=0}^{C_j-\mathbf{d}\cdot(\mathbf{y}\oplus(\mathbf{e}_i 1_{j\in\mathcal{R}_u}))} q_j(z) f(\mathbf{y}). \tag{19}$$

Here Z_j refers to the part of the capacity of link j that is occupied by non-multicast traffic.

To justify Eq. (19), consider two new sets for the collection of random variables $(\mathbf{Y}, Z)$, for $j \in \mathcal{J}$ and $i \in \mathcal{I}$,

$$\hat{\mathcal{S}}_j = \{(\mathbf{y}, z) \in \mathcal{S}\times\mathbb{N} | \mathbf{d}\cdot\mathbf{y} + z \le C_j\}, \qquad \hat{\mathcal{S}}_j^{u,i} = \{(\mathbf{y}, z) \in \mathcal{S}\times\mathbb{N} | \mathbf{d}\cdot(\mathbf{y}\oplus(\mathbf{e}_i 1_{j\in\mathcal{R}_u})) + z \le C_j\},$$

where ℕ denotes the set of natural numbers. Then, the probabilities $\hat{Q}_j$ and $\hat{Q}_j^{u,i}$ are expressed as

$$\hat{Q}_j(\mathbf{y}) = P(\mathbf{Y}_j = \mathbf{y}; (\mathbf{Y}_k, Z_k) \in \hat{\mathcal{S}}_k, \forall k \in \mathcal{M}_j) = P(\mathbf{Y}_j = \mathbf{y}, Z_j \leq C_j - \mathbf{d} \cdot \mathbf{y};$$
$$(\mathbf{Y}_k, Z_k) \in \hat{\mathcal{S}}_k, \forall k \in \mathcal{M}_j \setminus \{j\}) = P(Z_j \leq C_j - \mathbf{d} \cdot \mathbf{y}) \times P(\mathbf{Y}_j = \mathbf{y};$$
$$(\mathbf{Y}_k, Z_k) \in \hat{\mathcal{S}}_k, \forall k \in \mathcal{M}_j \setminus \{j\})$$

and

$$\begin{aligned}\hat{Q}_j^{u,i}(\mathbf{y}) &= P(\mathbf{Y}_j = \mathbf{y}; (\mathbf{Y}_k, Z_k) \in \hat{\mathcal{S}}_j^{u,i}, \forall k \in \mathcal{M}_j) \\ &= P(Z_j \leq C_j - \mathbf{d} \cdot (\mathbf{y} \oplus (\mathbf{e}_i 1_{j \in \mathcal{R}_u}))) \times P(\mathbf{Y}_j = \mathbf{y}; (\mathbf{Y}_k, Z_k) \in \hat{\mathcal{S}}_k, \forall k \in \mathcal{M}_j \setminus \{j\}).\end{aligned}$$

Thus, the algorithm differs only by the truncation function used

$$\hat{Q}_j(\mathbf{y}) = \begin{cases} \hat{T}_j \pi_j(\mathbf{y}) & \text{if } j \in \mathcal{U}, \\ \hat{T}_j \left[\bigotimes_{k \in \mathcal{N}_j} \hat{Q}_k \right] (\mathbf{y}) & \text{otherwise.} \end{cases}$$

Similarly,

$$\hat{Q}_j^{u,i}(\mathbf{y}) = \begin{cases} \hat{T}_j^{u,i} \pi_j(\mathbf{y}) & \text{if } j \in \mathcal{U}, \\ \hat{T}_j^{u,i} \left[\bigotimes_{k \in \mathcal{N}_j} \hat{Q}_k^{u,i} \right] (\mathbf{y}) & \text{otherwise.} \end{cases}$$

Another way of describing the relationship between the two different types of traffic, is to consider them as two traffic classes in a two-dimensional link occupancy state space as shown in Fig. 5. If the capacity is infinite, the traffic classes are independent of each other. The finite capacity of the link imposes a linear constraint for this state space. We notice that the marginal distribution of the capacity occupancy

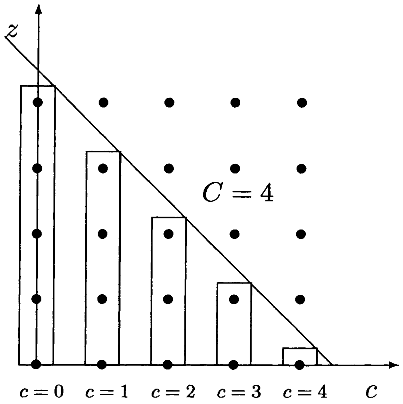

Fig. 5. Shaping of the marginal distribution of the capacity occupancy.

of the multicast traffic is weighted by the sums over the columns of the occupancy probabilities of the background traffic. If the multicast traffic occupies $c = \mathbf{d} \cdot \mathbf{y}_j$ units of capacity, and the link capacity is C_j, then possible non-multicast traffic states on the link are those with $z_j \leq C_j - c$, where z_j is the number of non-multicast calls, in accordance with Eq. (19).

The blocking probability in Eq. (12) is again obtained by two series of convolutions and truncations from the leaf links to the link J. The time blocking probability of the network is

$$\hat{b}^{\mathrm{t}}_{u,i} = 1 - \frac{\sum_{\mathbf{y}\in\mathcal{S}} \hat{Q}^{u,i}_J(\mathbf{y})}{\sum_{\mathbf{y}\in\mathcal{S}} \hat{Q}_J(\mathbf{y})}.$$

Recall that only the truncation operators used in the algorithm were altered. Therefore, the same modifications that were presented in Section 6 apply for the above algorithm.

8. Conclusions

We have presented an algorithm for calculating end-to-end time blocking probabilities in multicast networks exactly. The algorithm is based on the known algorithm for calculating blocking probabilities in hierarchical multiservice access networks. The multicast traffic characteristics were taken into account in the convolution step of the algorithm by introducing the new OR-convolution. As the complexity of the algorithm grows only linearly with the number of users (there is one OR-convolution and truncation operation per each added user), it makes an exact analysis of even large networks tractable, notwithstanding the fact that the size of the state space of the system grows exponentially with the number of users.

The algorithm was further extended to include background traffic in addition to multicast traffic. In the present paper the background traffic is assumed to consist of calls with capacity reservations, such as calls in an ATM network, or streaming type traffic with capacity reservations in the Internet. We have given four different user models satisfying the requirements for the use of the algorithm. The single user model presented in Section 4.1 can be considered as the main model, as from it the other three user population models can be derived. The main model can be modified in order to obtain the connection-specific user model presented by Chan and Geraniotis [2], and user models for arbitrary sized user populations.

We also showed how the original algorithm for calculating time blocking probabilities can be applied to calculating call blocking probabilities for all the user models presented. The results were further proven to be insensitive to the channel holding time distributions. For the finite user population models, the results for the time blocking probabilities were insensitive to user idle time distributions as well.

Calculating the end-to-end call blocking probability exactly, however, becomes infeasible when the number of channels increases. In contrast to ordinary access networks, the aggregate one dimensional link occupancy description is not sufficient, since in the multicast network it is essential to do all calculations in the link state space, with 2^I states. This is due to the resource sharing property of multicast traffic, namely the capacity in use on a link increases only when such a channel is requested which currently is not carried in the link. Thus, in cases where the number of channels is large, say more than 10, approximation methods such as RLA are needed.

We leave for further research extending the presented user population models to cover an even larger variety of realistic multicast user models and new approximation methods for calculating blocking probabilities. The acceleration of the presented algorithm for systems with a large number of channels should also be investigated. The complexity of the algorithm would decrease considerably if the calculations would not require the 2^I states of link state space. One possibility would be to assume that the channels be identical in terms of the probabilities of choosing the channels, the capacity requirement of the connection and the mean holding time of the receiver. These assumptions are clearly restrictive, but would allow the use of the algorithm for networks with many channels. Finally, an interesting area for further study is the adaptation of the present algorithm to the case where the background traffic depends on the state of the multicast transmissions, as in the case of elastic sources responding to the available bandwidth in the Internet.

Acknowledgements

We thank the anonymous referees for new ideas and suggestions for improving the paper.

Appendix A. On the insensitivity of multicast loss systems

In this appendix, we give a rigorous treatment of the insensitivity property discussed earlier. This is done by using the theory of generalized semi-Markov processes. The aim is to prove that the steady-state distribution $\tilde{\pi}(\mathbf{x})$ in a network with *arbitrary* link capacities is *insensitive* to the underlying connection holding time distributions, i.e. $\tilde{\pi}(\mathbf{x})$ depends only on the mean, but not on the form, of the connection holding time distributions. For the finite user population models, $\tilde{\pi}(\mathbf{x})$ proves out to be insensitive even to the user idle time distributions. Below, we will show that these properties are valid at least among the distribution classes defined in [11]. For shortness, these distributions are called general.

A.1. Finite user population models

The reversible Markov processes used to model finite user populations in Sections 4.1–4.3 implicitly require that all the user idle time and connection holding time distributions be exponential. If these exponential distributions are replaced by general distributions, without modifying their means, the resulting model is a semi-Markov process. It is well known that this semi-Markov process has the same stationary distribution as the original Markov process. Thus, by (3), we see that the steady-state distribution $\pi(\mathbf{x})$ in a network with *infinite* link capacities is insensitive to both the user idle time and connection holding time distributions. However, this does not prove the insensitivity of $\tilde{\pi}(\mathbf{x})$. Thus, a different approach is needed.

Consider first the user model defined in Section 4.1. If the user idle time and connection holding time distributions are exponential, the network state process $\tilde{\mathbf{X}}(t) = (\tilde{Y}_{ui}(t); u \in \mathcal{U}, i \in \mathcal{I})$ is a reversible Markov process satisfying the following *detailed balance equations*: for all $u \in \mathcal{U}$, $i \in \mathcal{I}$, and $\mathbf{x} \in \tilde{\Omega}$, such that $\mathbf{x} + \mathbf{e}_{ui} \in \tilde{\Omega}$,

$$\tilde{\pi}(\mathbf{x})\lambda_{ui} = \tilde{\pi}(\mathbf{x} + \mathbf{e}_{ui})\mu_i. \tag{A.1}$$

Assume then that the user idle time and connection holding time distributions are general. It is further assumed that whenever a connection request is blocked, the related user starts a fresh idle period. In this case, the network state process $\tilde{\mathbf{X}}(t)$, in general, is neither a Markov process nor a semi-Markov process.

Instead, it is a *generalized semi-Markov process* (GSMP) as defined, e.g., in [11]. The idea is to consider, in addition to the network state variable $\tilde{\mathbf{X}}(t)$, the remaining times $T_u(t)$ that each user u stays in *its* current state. If $\tilde{Y}_{ui}(t) = 1$, then $T_u(t)$ tells how long user u still continues to subscribe to channel i at time t (and we say that *clock* s_{ui} is *active* at that time). But if $\tilde{Y}_{ui}(t) = 0$ for all i, then $T_u(t)$ tells how long user u still remains idle at time t (and we say that clock s_{u0} is active at that time). Thus, $U(I+1)$ different clocks are needed, exactly U of which are active in each network state. Let then $\mathbf{T}(t) = (T_u(t); u \in \mathcal{U})$. The point is that the *supplemented* process $(\tilde{\mathbf{X}}(t), \mathbf{T}(t))$ is a Markov process.

By Theorem 1.1 of [11], the GSMP $\tilde{\mathbf{X}}(t)$ is insensitive to the user idle time and connection holding time distributions if the following *local balance equations* are satisfied:

(i) for all $u \in \mathcal{U}$ and $\mathbf{x} \in \tilde{\Omega}$ such that clock s_{u0} is active,

$$\tilde{\pi}(\mathbf{x})\lambda_u = \sum_{i\in\mathcal{I}} \tilde{\pi}(\mathbf{x}+\mathbf{e}_{ui})\mu_i 1_{\mathbf{x}+\mathbf{e}_{ui}\in\tilde{\Omega}} + \sum_{i\in\mathcal{I}} \tilde{\pi}(\mathbf{x})\lambda_{ui} 1_{\mathbf{x}+\mathbf{e}_{ui}\notin\tilde{\Omega}},$$

(ii) for all $u \in \mathcal{U}$, $i \in \mathcal{I}$, and $\mathbf{x} \in \tilde{\Omega}$ such that clock s_{ui} is active,

$$\tilde{\pi}(\mathbf{x})\mu_i = \tilde{\pi}(\mathbf{x}-\mathbf{e}_{ui})\lambda_{ui}.$$

It is easy to see that all these local balance equations follow from the detailed balance equation (A.1).

The user model defined in Section 4.2 can be handled similarly. In this case, the detailed balance equations are exactly the same ones (A.1) as above. The number of clocks needed is $2UI$: clock s_{uix} is active whenever $\tilde{Y}_{ui} = x \in \{0, 1\}$. The local balance equations corresponding to the insensitivity property read now as follows:

(i) for all $u \in \mathcal{U}$, $i \in \mathcal{I}$, and $\mathbf{x} \in \tilde{\Omega}$ such that clock s_{ui0} is active,

$$\tilde{\pi}(\mathbf{x})\lambda_{ui} = \tilde{\pi}(\mathbf{x}+\mathbf{e}_{ui})\mu_i 1_{\mathbf{x}+\mathbf{e}_{ui}\in\tilde{\Omega}} + \tilde{\pi}(\mathbf{x})\lambda_{ui} 1_{\mathbf{x}+\mathbf{e}_{ui}\notin\tilde{\Omega}},$$

(ii) for all $u \in \mathcal{U}$, $i \in \mathcal{I}$, and $\mathbf{x} \in \tilde{\Omega}$ such that clock s_{ui1} is active

$$\tilde{\pi}(\mathbf{x})\mu_i = \tilde{\pi}(\mathbf{x}-\mathbf{e}_{ui})\lambda_{ui}.$$

It is again easy to see that all these local balance equations follow from the detailed balance equations.

Consider finally the user population model defined in Section 4.3. The claim follows now from the first case above, Eq. (8) and the observation that the pure convolution operator preserves the insensitivity property.

A.2. Infinite user population model

The infinite user population was modeled in Section 4.4 as a collection of independent $M/M/\infty$ queues, requiring that all the interarrival time and connection holding time distributions be exponential. It is well known that the steady-state distribution in an $M/G/\infty$ queue is the same as in the corresponding $M/M/\infty$ queue (see, e.g., [5]). Thus, by (3), we see that $\pi(\mathbf{x})$ is insensitive to the connection holding time distributions. To prove the insensitivity of $\tilde{\pi}(\mathbf{x})$, we again apply the GSMP theory.

Consider first the extended network state process $\tilde{\mathbf{N}}(t) = (\tilde{N}_{ui}(t); u \in \mathcal{U}, i \in \mathcal{I})$, where $\tilde{N}_{ui}(t)$ denotes the number of ongoing multicast connections belonging to traffic class (u, i). Note that $\tilde{\mathbf{X}}(t) = \mathbf{h}(\tilde{\mathbf{N}}(t))$, where $\mathbf{h}(\mathbf{n}) = (1_{n_{ui}>0}; u \in \mathcal{U}, i \in \mathcal{I})$, so that $\tilde{Y}_{ui}(t) = 1_{\tilde{N}_{ui}(t)>0}$. Let then $\mathcal{E} = \{0, 1, \dots\}^{U\times I}$. The state space of $\tilde{\mathbf{N}}(t)$ is denoted by $\tilde{\mathcal{E}}$,

$$\tilde{\mathcal{E}} = \{\mathbf{n} \in \mathcal{E} | \mathbf{h}(\mathbf{n}) \in \tilde{\Omega}\}.$$

Let $\tilde{\pi}(\mathbf{n})$ denote the steady-state distribution of $\tilde{\mathbf{N}}(t)$. If the connection holding time distributions are exponential, the extended network state process $\tilde{\mathbf{N}}(t)$ is a reversible Markov process satisfying the following detailed balance equations: for all $u \in \mathcal{U}$, $i \in \mathcal{I}$, and $\mathbf{n} \in \tilde{\mathcal{E}}$, such that $\mathbf{n} + \mathbf{e}_{ui} \in \tilde{\mathcal{E}}$,

$$\tilde{\pi}(\mathbf{n})\lambda_{ui} = \tilde{\pi}(\mathbf{n} + \mathbf{e}_{ui})(n_{ui} + 1)\mu_i. \tag{A.2}$$

Assume then that the connection holding time distributions are general. In this case, basically due to an infinite user population, the extended network state process $\tilde{\mathbf{N}}(t)$ is not an ordinary GSMP but a GSMP resulting from a generalized semi-Markov scheme *with relabeling*, as defined in [11]. (A similar relabeling scheme is also needed when modeling an $M/G/\infty$ queue in this framework.) In addition to single clocks, different *clock types* are defined. In our case, there are $2UI$ different clock types: one clock of type S_{ui0} corresponding to the interarrival times is always active, and n clocks of type S_{ui1} corresponding to the holding times are active whenever $\tilde{N}_{ui}(t) = n$.

By Theorem 3.1 of [11], the GSMP $\tilde{\mathbf{N}}(t)$ resulting from the relabeling scheme described above is insensitive to the connection holding time distributions if the following local balance equations are satisfied: for all $u \in \mathcal{U}$, $i \in \mathcal{I}$, and $\mathbf{n} \in \tilde{\mathcal{E}}$, such that at least one of the clocks of type S_{ui1} is active,

$$\tilde{\pi}(\mathbf{n})\mu_i = \tilde{\pi}(\mathbf{n} - \mathbf{e}_{ui})\lambda_{ui}\frac{1}{n_{ui}}.$$

It is easy to see that all these local balance equations follow from the detailed balance equation (A.2). The insensitivity of $\tilde{\pi}(\mathbf{x})$ follows immediately from the insensitivity of $\tilde{\pi}(\mathbf{n})$.

References

[1] K. Bousseta, A.-L. Beylot, Multirate resource sharing for unicast and multicast connections, in: Proceedings of the Broadband Communications’99, 1999, pp. 561–570.

[2] W.C. Chan, E. Geraniotis, Tradeoff between blocking and dropping in multicasting networks, in: IEEE ICC’96 Conference Record, Vol. 2, 1996, pp. 1030–1034.

[3] J. Karvo, J. Virtamo, S. Aalto, O. Martikainen, Blocking of dynamic multicast connections in a single link, in: Proceedings of the Broadband Communications’98, 1998, pp. 473–483.

[4] J. Karvo, J. Virtamo, S. Aalto, O. Martikainen, Blocking of dynamic multicast connections, Telecommun. Syst. 16 (3–4) (2001) 467–481.

[5] F.P. Kelly, Reversibility and Stochastic Networks, Wiley, New York, 1979.

[6] C.K. Kim, Blocking probability of heterogeneous traffic in a multirate multicast switch, IEEE J. Selected Areas Commun. 14 (2) (1996) 374–385.

[7] M. Listanti, L. Veltri, Blocking probability of three-stage multicast switches, in: IEEE ICC’98 Conference Record, 1998, pp. 623–629.

[8] E. Nyberg, J. Virtamo, S. Aalto, An exact algorithm for calculating blocking probabilities in multicast networks, in: Proceedings of the Networking 2000, 2000, pp. 275–286.

[9] K.W. Ross, Multiservice Loss Models for Broadband Telecommunication Networks, Springer, London, 1995.

[10] N. Shacham, H. Yokota, Admission control algorithms for multicast sessions with multiple streams, IEEE J. Selected Areas Commun. 15 (3) (1997) 557–566.

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

[12] M. Stasiak, P. Zwierzykowski, Analytical model of ATM node with multicast switching, in: Proceedings of the Mediterranean Electrotechnical Conference, 1998, pp. 683–687.

[13] Y. Yang, J. Wang, On blocking probability of multicast networks, IEEE Trans. Commun. 46 (7) (1998) 957–968.

Eeva Nyberg received her M.Sc. (Tech.) degree in engineering mathematics from the Helsinki University of Technology in 1999. After her work on multicast loss systems, she is currently a Ph.D. student at the Networking Laboratory of the Helsinki University of Technology working on modeling and performance analysis of TCP traffic and DiffServ networks.

Jorma Virtamo received his M.Sc. (Tech.) degree in engineering physics and D.Sc. (Tech.) degree in theoretical physics from the Helsinki University of Technology, in 1970 and 1976, respectively. In 1986, he joined the Technical Research Centre of Finland, VTT Information Technology, where he led a Teletraffic Research Group, and became a research professor in 1995. Since 1997, he has been a professor at the Networking Laboratory of the Helsinki University of Technology. His current research interests include queuing theory and performance analysis and QoS mechanisms of the Internet, optical networks and ad hoc networks.

Samuli Aalto received the M.Sc. and Ph.D. degrees in mathematics from University of Helsinki in 1984 and 1998, respectively. From 1984 to 1997, he was with Technical Research Centre of Finland (VTT) as a research scientist in the area of telecommunications. Currently he is with Networking Laboratory of Helsinki University of Technology as an acting professor. Dr. Aalto's research areas are queuing and teletraffic theory with the current focus on teletraffic analysis of multicast loss systems and performance evaluation of DiffServ traffic control mechanisms.