MDP-based CAC and Routing in Loss-Delay Networks: Polynomial Cost Approximation

Ernst Nordström, Ming Fan

***Abstract*— In this paper we study the call admission control and routing issue in multi-service networks. Two categories of calls are considered: a narrow-band with blocked calls cleared and a wide-band with blocked calls delayed. The optimization is subject to several Quality of Service constraints, either on the packet or call level. The objective function is formulated as cost minimization with costs incurred by the rate of NB and WB reward loss and delay of WB calls. A suboptimal solution is achieved by applying Markov decision process (MDP) theory.**

I. INTRODUCTON

We consider the problem of optimal Call Admission Control (CAC) and routing in multi-service networks such as ATM and STM networks, and IP networks, provided they are extended with resource reservation capabilities. The objective is to maximize the revenue from carried calls, while meeting constraints on the Quality of Service (QoS) and Grade of Service (GoS) on the packet and call level, respectively.

The network is offered traffic from K call classes. Each call class is associated with one of P origin-destination (OD) node pairs. Each OD pair is offered traffic from G call categories, meaning that $K = PG$. For presentation simplicity, we assume $G = 2$ which is represented by one narrow-band (NB) category requesting a bandwidth of b_n Mbps, and one wide-band (WB) category requesting b_w Mbps $(b_n < b_w)$. The required bit is represented by the call's peak bandwidth in case of deterministic multiplexing, and by the call's equivalent bandwidth in case of statistical multiplexing.

This work was financially supported by the Swedish Knowledge Foundation.

E. Nordström is with Department of Culture, Media, Computer Sceince, Dalarna University, SE-781 88 Borlänge, Sweden

Ming Fan is with Department of Mathematics, Natural Sciences and Engineering, Dalarna University, SE-781 88 Borlänge, Sweden (e-mail: eno@du.se)

It is well known that when calls are set up on demand, the WB calls can suffer significantly higher rejection rates, compared to NB calls, if there is no additional mechanism to provide access fairness under overload conditions [16]. There exists two main approaches to cope with this fairness problem: access control of NB calls or queuing of WB calls.

Trunk reservation is a form of access control which reserves capacity to WB calls by rejecting NB calls when the link occupancy is over a threshold. While access control can deliver good fairness properties, this is usually achieved at the expense of bandwidth utilization.

Queuing of a WB call request is done when there is not sufficient free bandwidth to accept the call request. When a sufficient amount of bandwidth becomes available in the network, a waiting WB call is allowed to enter the network. This approach, if applied correctly, can provide access fairness and increased bandwidth utilization when compared with trunk reservation.

Modern CAC and routing mechanisms are state-dependent rather than static, which means that the decision to reject the request for a new call, or to accept it on a particular path depends on the current occupancy of the network. That is, the state of the network is represented by the number of calls from each class in service, or waiting for service, at each network link. A state-dependent CAC and routing policy is based on a mapping, for every call class, from a network state space to a set of possible routing decisions. First, CAC determines the set of feasible paths between the source and destination which offers sufficient QoS to the new and existing calls in terms of delay, delay variation, and data loss. Second, the network should select one path among the set of feasible paths to convey the call or to reject the request if call acceptance would diminish the expected revenue. While contributing to the maximization of the average revenue for the operator, this choice must comply with GoS constraints in terms of call blocking probabilities and call set-up delays. State-dependent mechanisms offer advantages both in terms of achievable revenue and ability to control the QoS and GoS.

This paper deals with a particular form of state-dependent CAC and routing, where the behavior of the network is formulated as Markov Decision Process (MDP) [6], [17]. A MDP is a controlled Markov process, where the set of state transitions from the current Markov state to other Markov states depends on the decision or action taken by the controller in the current state. Reward delivery from the user to the network can be modeled as occurring at call completion since this provides a correct model of carried reward.

Nordström and Dziong proposed a CAC and routing framework where blocked NB calls are cleared and blocked WB calls are delayed [11]. They formulated the control objective as maximization of a reward function being a linear combination of the reward from accepted NB and WB calls and the average WB call set-up delay treated as cost (penalty). A given OD pair can be offered traffic from several NB and WB categories which each can have an unique value of the reward parameter. Since both NB and WB calls are accounted for in the control objective it becomes possible to control the access fairness (call blocking probability) among both the NB and WB classes and not just among the NB classes as was suggested in [4]. The trade-off between NB and WB reward loss and average WB call set-up delay is controlled by the weight of the average delay term.

The computational burden of the exact MDP framework for CAC and routing is prohibitive even for moderate-size networks. Fortunately, it can be reduced to manageable levels by a set of modeling simplifications. First, the network is decomposed into a set of links assumed to have independent traffic and reward processes, respectively. Second, the K dimensional link Markov process and link reward process are aggregated into a G dimensional link Markov process and link reward process, respectively. Third, as will be studied in this paper, the exact G dimensional link MDP task is transformed into an approximate link MDP task which has reduced state space.

The computational burden of each link MDP task associated with the G dimensional link Markov process increases exponentially with the number of categories G. In order to cope with the problem imposed by large state spaces, several link MDP frameworks with reduced computational cost have been proposed, notably methods based on state aggregation [8], decomposition of the link Markov process [4], [13] and polynomial cost approximation [10], [15].

Krishnan and Hübner proposed a state aggregation link MDP framework for loss networks based on a scalar link state representing the link occupancy [8]. Transition probabilities between link states were derived from link occupancy probabilities obtained by a recursive procedure due to Kaufman [7] and Roberts [14]. The MDP task was solved by one-step policy iteration.

State aggregation can not be used between the NB loss category and the WB delay category, since the state space is not coordinate convex. The condition for coordinate convexity is not fulfilled since NB transitions are allowed from a given state (due to a NB call departure) but a transition in the other direction (due to a NB call arrival) is not always allowed. In order to maintain coordinate convexity the NB arrival should sometimes be able to preempt a WB from the link to the queue, which is not allowed.

The link Markov process decomposition method is due to Liao and Mason [9] and Dziong, Liao and Mason [4]. They observed that when the holding times of WB calls are significantly longer than for NB calls, the NB process changes state much more often than the WB process. This justifies that the NB and WB process can be analyzed separately. The NB process is analyzed separately for each state of the WB process, and the WB process is analyzed by taking the average ”disturbance” of the NB process into account. Nordström and Dziong proposed a link model based on Markov process decomposition for the CAC and routing objective adopted in this paper [13].

Marbach, Mihatch and Tsitsiklis applied reinforcement learning to estimate the optimal second-degree polynomial link-cost approximation [10]. Although the complexity of each simulation step is fixed and low, the required number of simulation steps is large (in the order of 10^7).

Rummukainen and Virtamo proposed an analytical loss link model for computing the the cost relative values as a linear combination of a modest number of basis vectors [15]. Single-coordinate and double-coordinate monomial vectors up to some degree, were considered as basis vectors. The MDP task was solved by one-step policy iteration. The accuracy of the proposed polynomial cost approximation was only evaluated on the link level for a loss network.

The aim of this paper is twofold. First, we extend Rummukainen's and Virtamo's MDP framework to mixed loss-delay networks. In particular, we formulate a link model for the polynomial cost approximation, and a numerical procedure for setting up the equations in this link model. Second, we present a numerical simulation-based performance evaluation on the network level of polynomial cost approximation. We compare the results with the exact link MDP model and the link models based on state aggregation [8] and decomposition of the link Markov process [13]. We also compare with conventional routing represented by the Least Loaded Routing (LLR) method.

The paper is organized as follows: Section II formulates the CAC and routing problem in terms of offered traffic, network model, and optimization objective. Section III describes the network decomposition and the exact link MDP model. Section IV outlines the CAC and routing framework. Section V proposes a new loss-delay link MDP model for polynomial cost approximation. Section VI presents the simulation results for MDP routing based on exact and approximate link models, and for LLR routing. Finally, Section VII concludes the paper.

II. CAC AND ROUTING PROBLEM FORMULATION

A. *Traffic assumptions and optimization objective*

The network is offered traffic from K classes which are, for sake of simplicity, subject to deterministic multiplexing. The j-th class, $j \in J = \{1, \ldots, K\}$, is characterized by the following:

- Origin-destination (OD) node pair,
- bandwidth requirement b_j [Mbps],
- Poisson arrival process with rate λ_j [s^{-1}],
- Exponentially distributed holding time with mean $1/\mu_j$ [s],
- Set of alternative routes, W_j, and
- Reward parameter $r_j \in (0, \infty)$.

The parameter r_j is a CAC and routing control parameter that can be used to achieve several different objectives of the network operator. In particular it can be used to maximize the network revenue if the reward parameters are proportional to the call charging. It can also be used to achieve fairness in network access by increasing the reward parameters for handicapped calls and vice versa.

On the network links, the classes are aggregated into $G = 2$ bandwidth categories. The i-th category, $i \in I = \{1, 2\} = \{n, w\}$, on link s, is characterized by:

- bandwidth requirement $b_i \in \{b_n, b_w\}$ [Mbps],
- Average mean call holding time $1/\overline{\mu}_i^s$ [s],
- Average reward parameter $\overline{r}_i^s(\pi)$.

where π denotes the CAC and routing policy. See [11] on details on how to compute $\overline{\mu}_i^s$ and $\overline{r}_i^s(\pi)$.

The objective function is formulated as cost minimization with costs incurred by the rate of reward loss due to blocking of NB and WB calls and delay of WB calls:

$$\overline{W}_D = \overline{W} + \sum_s \alpha^s \overline{D}^s \frac{\lambda_w^s}{\lambda_w} \quad (1)$$

where λ_w^s, λ_w denotes the arrival rate of WB calls offered to the s-th link and totally, to all OD pairs of the network, respectively, and $\overline{D}^s$ denotes the average WB call-set up delay for link s, α^s denotes the weight of the delay term for link s, and $\overline{W}$ denotes the rate of lost reward:

$$\overline{W} = \sum_j r_j \lambda_j B_j \tag{2}$$

where B_j denotes the j-th class call blocking probability.

B. Network and queuing model

The network is assumed to consist of a set of switching nodes. The switching nodes communicate in both traffic flow directions using uni-directional links. Each uni-directional link has one finite FIFO queue for WB call requests. The following basic scheme of queuing system management, originally proposed in [4], is assumed throughout the paper. When the path chosen by the CAC and routing algorithm has sufficient available capacity for the new WB call, the call is set up between the considered origin-destination node. Otherwise, at least one link along the path is not able to directly accept the new WB call. At those links, the new WB call request joins the queue at the tail. We assume that the path would not be chosen when some of its links has insufficient capacity on both the link and in the queue. At links with sufficient capacity, bandwidth is reserved for the new WB call while waiting for all links to be ready to accept the call. A link queue is served when a sufficient number or bandwidth units becomes available on the link. In this case, bandwidth for the WB call at the head of the queue is reserved on the link. When bandwidth has been reserved on every link along the path for a given WB call, the call is set up between the considered origin-destination node.

III. MDP MODELING

A. Network decomposition

The behavior of the network under consideration can be described by a Markov Decision Process (MDP) with the objective to minimize the cost function defined by (1). The network action space is given by

$$A = \{\mathbf{a} = \{a_j\} : a_j \in \{0\} \cup W_j, j \in J\}, \tag{3}$$

where $a_j = 0$ denotes call rejection and the set W_j contains the indices of the alternative routes possible for an accepted class j call.

The network cost rate, $\rho(\mathbf{z})$, is given by:

$$\rho(\mathbf{z}) = \sum_{j \in J} r_j \lambda_j (1 - \theta(a_j)) + \sum_{s \in S} \beta^s \frac{z_l^s}{\lambda_w}, \tag{4}$$

where $\mathbf{z}$ denotes the network state and z_j, z_l^s denote the number of the j-th type calls and the number of calls in the s-th queue in state $\mathbf{z}$, respectively, θ denotes the indicator function which is one for positive arguments and zero otherwise, and S denotes the set of all link indices in the network.

The network state and action spaces can be very large, even for moderate-size networks. We therefore decompose the network into a set of links assumed to have independent traffic and reward processes, respectively [5].

The network Markov process is decomposed into a set of independent link Markov processes, driven by state-dependent Poisson call arrival processes with rate $\lambda_j^s(\mathbf{x}, \pi)$. In particular, a call connected on a path consisting of l links is decomposed into l independent link calls characterized by the same mean call holding time as the original call.

The network reward process is decomposed into a set of separable link reward processes. The link call reward parameters $r_j^s(\pi)$ fulfill the obvious condition that

$$r_j = \sum_{s \in S_k} r_j^s(\pi), \tag{5}$$

where S_k denotes the set of links constituting path k, specified by the routing policy π. Different models for computing link reward parameters are possible [5]. In this paper we use a simple rule: the call reward is distributed uniformly among the path's links, resulting in the formula $r_j^s(\pi) = r_j / l$, where l denotes the number links in the call's path.

Even in the decomposed network model, the state space can be quite large when many call classes share the links. One way to reduce the state space is to construct a modified link reward process in which the link call *classes* with the same bandwidth requirement are aggregated into one *category* $i \in I$ with average reward parameter defined as [5]:

$$\overline{r}_i^s(\pi) = \frac{\sum_{j\in J_i} r_j^s(\pi)\overline{\lambda}_j^s(\pi)}{\sum_{j\in J_i} \overline{\lambda}_j^s(\pi)}, \quad (6)$$

where J_i denotes the set of classes that belongs to the i-th category, and $\overline{\lambda}_j^s(\pi)$ denotes the average rate of class-j calls accepted on link s. In the following, this simplification is adopted, which reduces the number of effective classes to the number of classes with unique bandwidth requirement.

B. Exact link model

The semi-Markov decision process (SMDP) model for the link s consists of:

- Ω^s is a finite set of states,
- A^s is a finite set of actions,
- $q^s : \Omega \times \Omega \times A \to \mathbf{R}$ is the infinitesmal generator matrix of transition rates between states,
- $\rho^s : \Omega \times A \to \mathbf{R}$ is the immediate cost rate function for taking different actions in each state.

The state in the exact link model is given by $\mathbf{x} = (x_n, x'_w)^T$, where x_i denotes the number of category i calls on the link, and x'_i denotes the number of category i calls in the system (on the link and, possibly, in the queue). The state space Ω^s for the exact link model is given by:

$$\begin{aligned}\Omega^s = \quad & \Big\{\mathbf{x} = (x_n, x'_w)^T : 0 \le x_n \le N_n^s, 0 \le x'_w \le \tilde{N}_w^s, \\ & x_l = f_l(\mathbf{x}), x_n b_n + (x'_w - x_l) b_w \le C^s\Big\},\end{aligned}$$

where $N_n^s = \lfloor C^s/b_n \rfloor$, $\tilde{N}_w^s = \lfloor C^s/b_w \rfloor + L^s$, and C^s, L^s, denotes the capacity and maximal size of link and queue s, respectively. For convenience, we denote by $\Omega^s(c)$ the subset of states where exactly c trunks are occupied:

$$\Omega^s(c) = \left\{\mathbf{x} = (x_n, x'_w)^T \in \Omega^s | \mathbf{x}^T\mathbf{b} = c\right\}, \qquad c = 0\ldots, \tilde{C}^s. \quad (8)$$

where $\tilde{C}^s = C^s + b_w L^s$ denotes the system capacity, and $\mathbf{b} = (b_n, b_w)^T$ denotes the vector with bandwidth requirements. The cardinality of the state space is denoted by N. The number of WB calls in the queue, x_l, is obtained from state $\mathbf{x}$ as follows:

$$x_l = f_l(\mathbf{x}) := \inf\{x_l : x_l \ge 0, C^s - x_n b_n \ge ({x_w}' - x_l) b_w\}. \quad (9)$$

The MDP action is represented by a vector $\mathbf{a} = \{a_i\}, i \in I$, corresponding to admission decisions for presumptive call requests. The action space is given by:

$$A^s = \{\mathbf{a} = \{a_i\} : a_i \in \{0, 1\}, i \in I\}, \quad (10)$$

where $a_i = 0$ denotes call rejection and $a_i = 1$ denotes call acceptance. The permissible action space is a state-dependent subset of A:

$$A^s(\mathbf{x}) = \{\mathbf{a} \in A^s : a_i = 0 \text{ if } \mathbf{x} + \mathbf{e}_i \notin \Omega^s, i \in I\} \quad (11)$$

where $\mathbf{e}_i$ denotes a vector of zeros except a one in position $i \in I$.

The N by N infinitesmal generator matrix Q of the link-state process under CAC and routing policy π is defined by

$$\begin{aligned} & q^s_{\mathbf{xy}}(\mathbf{a}) \\ & = \begin{cases} \lambda_i^s(\mathbf{x}, \pi) a_i & \mathbf{y} = \mathbf{x} + \mathbf{e}_i \in \Omega^s, \\ x_i \overline{\mu}_i & \mathbf{y} = \mathbf{x} - \mathbf{e}_i \in \Omega^s, \\ -(\sum_{i\in I} x_i\overline{\mu}_i + \lambda_i^s(\mathbf{x}, \pi) a_i) & \mathbf{y} = \mathbf{x} \in \Omega^s, \\ 0, & \text{otherwise}, \end{cases} \end{aligned} \quad (12)$$

where $\lambda_i^s(\mathbf{x}, \pi)$ denotes the i-th category arrival rate to the link in state $\mathbf{x}$, see [5] for details of computation.

The expected cost rate in state $\mathbf{x}$ is given by $\rho^s(\mathbf{x}, \mathbf{a})$:

$$\rho^s(\mathbf{x}, \mathbf{a}) = \sum_{i\in I} (1 - a_i) \overline{r}_i^s \lambda_i^s(\mathbf{x}, \pi) + \alpha^s \frac{x_l}{\lambda_w}, \quad (13)$$

IV. CAC AND ROUTING FRAMEWORK

This section outlines the MDP computational procedure for determining a near-optimal CAC and routing policy using the exact link model. The central idea is to compute *path net-gain* functions, $g_j^k(\mathbf{y}, \pi)$, which estimate the increase in long-term reward due to admission of a class j call on path k in network state $\mathbf{y}$. The CAC and routing rule is simply to choose, given the state of the network and the class of the call request, a path which offers maximal positive path net-gain among the paths with sufficient QoS. The call is rejected if the path net-gain is negative, or if no path would offer sufficient QoS.

A. Basic definitions

The state-dependent path net-gain is defined as:

$$g_j^k(\mathbf{y},\pi) = r_j - \sum_{s\in S_k} p_i^s(\mathbf{x},\pi), \tag{14}$$

where $\mathbf{y} = \{\mathbf{x}\}$ denotes the network state in the decomposed network model. The link shadow price $p_i^s(\mathbf{x},\pi)$ can be interpreted as the expected cost for accepting an i-th category call in state $\mathbf{x}$. The link shadow price is defined as follows:

$$p_i^s(\mathbf{x},\pi) = v^s(\mathbf{x}+\mathbf{e}_i,\pi) - v^s(\mathbf{x},\pi). \tag{15}$$

where $v^s(\mathbf{x},\pi)$ denotes the *relative value* for category i in state $\mathbf{x}$. The relative value in state $\mathbf{x}$ is defined as the difference in future cost incurments when starting in the given state, compared to a reference state, $\mathbf{x}_r$. In practice, the relative value function is obtained by solving a set of linear equations (see Section IV-B).

B. Adaptation of the CAC and routing policy

The algorithm for determining the near-optimal CAC and routing policy π can be summarized as follows:

1) **Startup**: Initialize the relative values $v^s(\mathbf{x},\pi)$ in a way that make all link net-gains with permissible admission positive.
2) **On-line operation phase**: measure per-path call acceptance rates $\overline{\lambda}_j^k(\pi)$ and per-link blocking probabilities $B_j^s(\pi)$ while employing the maximum path net-gain routing rule. Perform the measurements for a sufficiently long period for the system to attain statistical equilibrium.
3) **Policy iteration cycle**: At the end of the measurement period, perform the following steps for all links s in the network:
   a) **Identify the link MDP model**: Determine per-category reward parameters $\overline{r}_i^s(\pi)$ and link call arrival rates $\lambda_i^s(\mathbf{x},\pi)$.
   b) **Value determination**: Find the relative values $v^s(\mathbf{x},\pi)$ and average cost rate $\overline{W}^s(\pi)$ for the current routing policy π.
   c) **Policy improvement**: Find the new link CAC policies π_s' based on the new relative values and the new average cost rate.
4) **Convergence test**: Repeat from 2 until average cost per time unit converges.

According to MDP theory an optimal policy is found after a finite number of policy iterations in case of a finite state and policy space [17].

1) Value determination: The value determination step for link s determines the relative values $v^s(\mathbf{x},\pi)$ for all states $\mathbf{x}\in\Omega^s$ and the average cost rate $\overline{W}^s(\pi)$ by solving a sparse system of linear Howard equations:

$$\begin{cases} v^s(\mathbf{x},\pi) = \rho^s(\mathbf{x},\mathbf{a}) - \overline{W}^s(\pi) + \sum_{\mathbf{y}\in\Omega^s} q^s_{\mathbf{xy}}(\mathbf{a}) v^s(\mathbf{y},\pi) \\ v^s(\mathbf{x}_r,\pi) = 0; \qquad \mathbf{x}\in\Omega^s \setminus \{\mathbf{x}_r\}. \end{cases} \tag{16}$$

The computation (time) complexity of the value determination step of policy iteration is a function of the size, N, of the state space. Traditional Gauss elimination has complexity O(N^3). This can be seen as an upper limit of the actual complexity since the system is sparse and more efficient iterative algorithms can be used.

2) Policy improvement: The policy improvement step for link s consists of finding the action that maximizes the relative value in each state $\mathbf{x}\in\Omega^s$:

$$\mathbf{a} = \underset{\mathbf{u}\in A^s(\mathbf{x})}{\operatorname{argmax}} \left\{ \rho^s(\mathbf{x},\mathbf{u}) - \overline{W}^s(\pi) + \sum_{y\in\Omega^s} q^s_{\mathbf{xy}}(\mathbf{u}) v^s(\mathbf{y},\pi) \right\}, \tag{17}$$

where $A^s(\mathbf{x})$ denotes the set of possible actions in state $\mathbf{x}$. The set of actions which yields the maximum improvement of relative values constitute an improved policy π_s' to be used again in the first step. The policy improvement step has complexity O($2^G N$), where G denotes the number of unique bandwidth categories.

V. POLYNOMIAL COST APPOXIMATION

A. Normal equations

Let $\mathbf{v}^s(\pi)$ denote the N-vector of relative values $v^s(\mathbf{x},\pi), \mathbf{x} \in \Omega^s$. In the polynomial cost approximation we express $\mathbf{v}^s(\pi)$ as a linear combination of a modest number of basis vectors $\mathbf{u}_h$, $h = 1,\dots,H$. In matrix form, we have

$$\mathbf{v}^s(\pi) = \sum_{h=1}^{H} \alpha_h(\pi)\mathbf{u}_h = U\alpha, \tag{18}$$

where $\alpha(\pi) = (\alpha_1(\pi),\dots,\alpha_H(\pi))^T$ are the free coefficients of the basis vectors, and U is a N by H matrix with vectors $\mathbf{u}_h$,$h = 1,\dots,H$ as columns. Chosing the empty state as reference state $\mathbf{x}_r$ means that we require

$$[\mathbf{u}_h]_{\mathbf{0}} = 0 \quad \text{for all } h = 1,\dots,H. \tag{19}$$

Substituting the parametric relative value representation (18) in the Howard equations (16) yields the overdetermined linear system of N equations in H variables:

$$\rho^s(\pi) - \overline{W}^s(\pi)\mathbf{1} + QU\alpha(\pi) = 0 \tag{20}$$

where $\rho^s(\pi)$ denotes the N-vector of immediate cost rates. Hence, we have a linear least-square problem were the task is to find the coefficient vector that minimizes the Euclidean norm of the left-hand side of (20).

According to standard theory for least-square problems, the coefficient vector $\alpha(\pi)$ minimizing the Euclidean norm of the left-hand side of (20) can be determined as the solution of the normal equations:

$$U^TQ^TQU\alpha(\pi) = U^TQ^T(\overline{W}^s(\pi)\mathbf{1} - \rho^s(\pi)) \tag{21}$$

This is a symmetric linear system of H equations in H variables.

The average cost rate $\overline{W}^s(\pi)$ can be treated as an unknown variable as follows. Let us rearrange the Howard equations (20) in the form

$$\begin{pmatrix} -1 & QU \end{pmatrix} \begin{pmatrix} \overline{W}^s(\pi) \\ \alpha(\pi) \end{pmatrix} = -\rho^{\mathbf{s}}(\pi) \tag{22}$$

Considering this as an overdetermined linear system of N equations in $H+1$ variables, the parameters that mimimize the Euclidean norm of the residual vector can be determined from the extended set of normal equations

$$\begin{pmatrix} N & -\mathbf{1}^TQU \\ -U^TQ^T\mathbf{1} & U^TQ^TQU \end{pmatrix} \begin{pmatrix} \overline{W}^s(\pi) \\ \alpha(\pi) \end{pmatrix} = \begin{pmatrix} \mathbf{1}^T \\ -U^TQ^T \end{pmatrix} \rho^s(\pi) \tag{23}$$

Polynomial cost approximation employs one-step policy iteration. This means that the policy improvement step is implicitly implemented by selecting the path with maximum path net-gain (no explicit policy improvement step is necessary for each link). Since the relative values for each link are less prone to change at each adaptation epoch, convergence occurs faster than for the exact link MDP model.

B. Basis vectors

The basis vectors were chosen as follows [15]. First, we consider the family of monomial basis vector $\mathbf{u}(\nu), \nu \in \mathbb{N}^G$, with vector elements defined by

$$[\mathbf{u}(\nu)]_{\mathbf{x}} = \prod_{i=1}^{G} {x'}_i^{\nu_i}, \quad \text{for } \mathbf{x} \in \Omega^s, \tag{24}$$

where 0^0 is taken 1 so that $\nu_i = 0$ indicates the the ith factor is always unity; this interpretation holds for all potential ocurrences of 0^0 in this paper. For simplicity, we assume that the exponent vector ν does not contain mote than two nonzero elements, thus restricting the discussion to the single-coordinate monomials ${x'}_i^{\nu_i}$ and the double-coordinate monomials ${x'}_i^{\nu_i}{x'}_k^{\nu_k}$. Note that (19) makes it unnecessary to consider the case $\nu = 0$ in which all elements of $\mathbf{u}(\mathbf{0})$ are equal to 1.

Second, we consider the piecewise monomial basis vectors $\mathbf{u}(\nu, d), \nu \in \mathbb{N}^G, d = 1,\dots,\tilde{C}^s$, with the vector elements defined by

$$[\mathbf{u}(\nu, d]_{\mathbf{x}} = 1_{\mathbf{x}\in\Omega^s(d)} \prod_{i=1}^{G} {x'}_i^{\nu_i}, \mathbf{x} \in \Omega^s, \tag{25}$$

Here, we require that ν to have at most one nonzero element, so that elements of these vectors are either single-coordinate piecewise monomials $1_{\mathbf{x}\in\Omega^s(d)}{x'}_i^{\nu_i}$ or piecewise constants $1_{\mathbf{x}\in\Omega^s(d)}$.

We specify the complete basis in terms of the integer parameters D_1, D_2, E_2 and P_1, where $E_2 \geq D_2$, as comprising the single-coordinate vectors up to degree D_1 [15]:

$$\mathbf{u}(\alpha\mathbf{e}_m), \quad \text{for } \alpha = P_1+1,\ldots,D_1; m \in I,$$

the double-coordinate monomial vectors up to degree D_2+E_2:

$$\begin{aligned}\mathbf{u}(\alpha\mathbf{e}_m+\beta\mathbf{e}_n),\ &\text{for } \alpha=1,\ldots,D_2; \beta=1,\ldots,E_2;\\ &m,n\in I, m\neq n,\end{aligned}$$

the piecewise constant vectors

$$\mathbf{u}(\mathbf{0},d), \quad \text{for } d=1,\ldots,\tilde{C}^s,$$

the piecewise single-coordinate mononimal vectors up to degree P_1

$$\begin{aligned}\mathbf{u}(\alpha\mathbf{e}_m,d), \quad &\text{for } \alpha=1,\ldots,P_1; m\in I;\\ &d=1,\ldots,\tilde{C}^s,\end{aligned}$$

the piecewise double-coordinate monomial basis vectors up to degree D_2+E_2

$$\begin{aligned}\mathbf{u}(\alpha\mathbf{e}_m+\beta\mathbf{e}_n,d),\ &\text{for } \alpha=1,\ldots,D_2; \beta=1,\ldots,E_2;\\ &m,n\in I, m\neq n, d=1,\ldots,\tilde{C}^s,\end{aligned}$$

the linear combinations of piecewise single-coordinate basis vectors

$$\begin{aligned}\mathbf{u}(\alpha\mathbf{e}_m,1,\tilde{C}^s-e) = \sum_{d=0}^{\tilde{C}^s-e}\mathbf{u}(\alpha\mathbf{e}_m,d),\\ \text{for } \alpha=1,\ldots,P1; m\in I.\end{aligned} \tag{26}$$

and the linear combinations of piecewise double-coordinate basis vectors

$$\begin{aligned}\mathbf{u}(\alpha\mathbf{e}_m+\beta\mathbf{e}_n,1,\tilde{C}^s-e) = \sum_{d=0}^{\tilde{C}^s-e}\mathbf{u}(\alpha\mathbf{e}_m+\beta\mathbf{e}_n,d),\\ \text{for } \alpha=1,\ldots,P1; \beta=1,\ldots,E_2; m\in I.\end{aligned} \tag{27}$$

The linear combination (26) replaces, for effiency reasons, the $\mathbf{u}(\alpha\mathbf{e}_m,d), d=1,\ldots,\tilde{C}^s-e$. The remaining vectors $\mathbf{u}(\alpha\mathbf{e}_m,d), d=\tilde{C}^s-e+1,\ldots,\tilde{C}^s$ are kept separate. Similary for the piecewise double-coordinte basis vectors.

The total number of basis vectors is now

$$\begin{aligned}H = \ &(D_1-P_1)+D_2(2E_2-D_2)\tfrac{1}{2}G(G-1)(e+2)+\\ &P_1G(e+1)+\tilde{C}^s.\end{aligned} \tag{28}$$

C. Structure of the normal equations

The elements of the right-hand side vector of the normal equations are given by

$$\begin{aligned}[U^TQ^T\mathbf{1}]_h = \ &\sum_{\mathbf{x}\in\Omega^s}[Q\mathbf{u}_h]_{\mathbf{x}}\\ &\text{for } h=1,\ldots,H,\end{aligned} \tag{29}$$

and the elements of the matrix U^TQ^TQU are

$$\begin{aligned}[U^TQ^TQU]_{hw} = \ &\sum_{\mathbf{x}\in\Omega^s}[Q\mathbf{u}_h]_{\mathbf{x}}[Q\mathbf{u}_w]_{\mathbf{x}}\\ &\text{for } h,w=1,\ldots,H.\end{aligned} \tag{30}$$

In order to further deconstruct the matrix structures, let us take advantage of the sparsity of Q defined by (12), so as to express an element of $Q\mathbf{u}_h$ as

$$\begin{aligned}[Q\mathbf{u}_h]_{\mathbf{x}} = \ &\sum_{i=1}^{G}a_i\lambda_i^s(\mathbf{x},\pi)([\mathbf{u}_h]_{\mathbf{x}+\mathbf{e}_i}-[\mathbf{u}_h]_{\mathbf{x}})+\\ &\sum_{i=1}^{G}x_i\overline{\mu}_i([\mathbf{u}_h]_{\mathbf{x}-\mathbf{e}_i}-[\mathbf{u}_h]_{\mathbf{x}})\end{aligned} \tag{31}$$

Let us now derive explicit forms for the elements of the vectors $Q\mathbf{u}(\nu)$ and $Q\mathbf{u}(\nu,d)$ where ν is restricted to the simple forms considered in Section V-B. To begin with, for single-coordinate monomial basis vectors $\mathbf{u}(\alpha\mathbf{e}_m), \alpha>0$, the elements differences in (31) can be expanded as

$$\begin{aligned}&[\mathbf{u}(\alpha\mathbf{e}_m)]_{\mathbf{x}\pm\mathbf{e}_i}-[\mathbf{u}(\alpha\mathbf{e}_m)]_{\mathbf{x}}\\ &=({x'}_m\pm 1_{i=m})^{\alpha}-{x'}_m^{\alpha}\\ &=1_{i=m}\sum_{\theta=0}^{\alpha-1}\binom{\alpha}{\theta}(\pm1)^{\alpha-\theta}{x'}_m^{\theta},\end{aligned} \tag{32}$$

We apply this expression to (31) for the various basis vectors. For single-coordinate monomial basis vectors $\mathbf{u}(\alpha\mathbf{e}_m), \alpha > 0$, we get

$$[Q\mathbf{u}(\alpha\mathbf{e}_m)]_{\mathbf{x}} = \sum_{i=1}^{G} a_i\lambda_i^s(\mathbf{x},\pi)1_{i=m}\sum_{\theta=0}^{\alpha-1}\binom{\alpha}{\theta}{x'}_m^{\theta} + \sum_{i=1}^{G} x_i\overline{\mu}_i 1_{i=m}\sum_{\theta=0}^{\alpha-1}\binom{\alpha}{\theta}(-1)^{\alpha-\theta}{x'}_m^{\theta} \quad (33)$$

For double-coordinate monomial basis vectors $\mathbf{u}(\alpha\mathbf{e}_m+\beta\mathbf{e}_n)$, where $m \neq n$ and $\alpha, \beta > 0$, we get

$$\begin{aligned}[Q\mathbf{u}(\alpha\mathbf{e}_m+\beta\mathbf{e}_n)]_{\mathbf{x}} &= \sum_{i=1}^{G} a_i\lambda_i^s(\mathbf{x},\pi)1_{i=m}\sum_{\theta=0}^{\alpha-1}\binom{\alpha}{\theta}{x'}_m^{\theta}{x'}_n^{\beta} \\ &+\sum_{i=1}^{G} a_i\lambda_i^s(\mathbf{x},\pi)1_{i=n}\sum_{\theta=0}^{\beta-1}\binom{\beta}{\theta}{x'}_m^{\alpha}{x'}_n^{\theta} \\ &+\sum_{i=1}^{G} x_i\overline{\mu}_i 1_{i=m}\sum_{\theta=0}^{\alpha-1}\binom{\alpha}{\theta}(-1)^{\alpha-\theta}{x'}_m^{\theta}{x'}_n^{\beta} \\ &+\sum_{i=1}^{G} x_i\overline{\mu}_i 1_{i=n}\sum_{\theta=0}^{\beta-1}\binom{\beta}{\theta}(-1)^{\beta-\theta}{x'}_m^{\alpha}{x'}_n^{\theta}\end{aligned} \quad (34)$$

For piecewise constant vectors $\mathbf{u}(\mathbf{0},d), d = 1,\dots,\tilde{C}^s$, we get

$$[Q\mathbf{u}(\mathbf{0},d)]_{\mathbf{x}} = \begin{cases} a_i\lambda_i^s(\mathbf{x},\pi), & \mathbf{x}^T\mathbf{b} = d - b_i \text{ for some } i \\ -\sum_{i=1}^{G}[a_i\lambda_i^s(\mathbf{x},\pi) + x_i\overline{\mu}_i], & \mathbf{x}^T\mathbf{b} = d \\ x_i\overline{\mu}_i, & \mathbf{x}^T\mathbf{b} = d + b_i \text{ for some } i \\ 0, & \text{otherwise} \end{cases} \quad (35)$$

For piecewise single monomial vectors $\mathbf{u}(\alpha\mathbf{e}_m,d), \alpha > 0, d = 1,\dots,\tilde{C}^s$, we get

$$[Q\mathbf{u}(\alpha\mathbf{e}_m,d)]_{\mathbf{x}} = \begin{cases} a_i\lambda_i^s(\mathbf{x},\pi){x'}_m^{\alpha}, & \mathbf{x}^T\mathbf{b} = d - b_i \text{ for } i \neq m \\ \sum_{\theta=0}^{\alpha} a_m\lambda_m^s(\mathbf{x},\pi)\binom{\alpha}{\theta}{x'}_m^{\theta}, & \mathbf{x}^T\mathbf{b} = d - b_m \\ -\sum_{i=1}^{G}[a_i\lambda_i^s(\mathbf{x},\pi)+x_i\overline{\mu}_i]{x'}_m^{\alpha}, & \mathbf{x}^T\mathbf{b} = d \\ \sum_{\theta=0}^{\alpha} x_m\overline{\mu}_m\binom{\alpha}{\theta}(-1)^{\alpha-\theta}{x'}_m^{\theta}, & \mathbf{x}^T\mathbf{b} = d + b_m \\ x_i\overline{\mu}_i{x'}_m^{\alpha}, & \mathbf{x}^T\mathbf{b} = d + b_i \text{ for } i \neq m \\ 0, & \text{otherwise.} \end{cases} \quad (36)$$

For piecewise double monomial vectors $\mathbf{u}(\alpha\mathbf{e}_m + \beta\mathbf{e}_n, d), \alpha, \beta > 0$, $d = 1,\dots,\tilde{C}^s$, we get

$$[Q\mathbf{u}(\alpha\mathbf{e}_m+\beta\mathbf{e}_n,d)]_{\mathbf{x}} = \begin{cases} a_i\lambda_i^s(\mathbf{x},\pi){x'}_m^{\alpha}{x'}_n^{\beta}, & \mathbf{x}^T\mathbf{b} = d - b_i \text{ for } i \neq m \\ \sum_{\theta=0}^{\alpha} a_m\lambda_m^s(\mathbf{x},\pi)\binom{\alpha}{\theta}{x'}_m^{\theta}{x'}_n^{\beta} + \sum_{\theta=0}^{\beta} a_n\lambda_n^s(\mathbf{x},\pi)\binom{\beta}{\theta}{x'}_m^{\alpha}{x'}_n^{\theta}, & \mathbf{x}^T\mathbf{b} = d - b_m \\ -\sum_{i=1}^{G}[a_i\lambda_i^s(\mathbf{x},\pi)+x_i\overline{\mu}_i]{x'}_m^{\alpha}{x'}_n^{\beta}, & \mathbf{x}^T\mathbf{b} = d \\ \sum_{\theta=0}^{\alpha} x_m\overline{\mu}_m\binom{\alpha}{\theta}(-1)^{\alpha-\theta}{x'}_m^{\theta}{x'}_n^{\beta} + \sum_{\theta=0}^{\beta} x_n\overline{\mu}_n\binom{\beta}{\theta}(-1)^{\beta-\theta}{x'}_m^{\alpha}{x'}_n^{\theta}, & \mathbf{x}^T\mathbf{b} = d + b_m \\ x_i\overline{\mu}_i{x'}_m^{\alpha}{x'}_n^{\beta}, & \mathbf{x}^T\mathbf{b} = d + b_i \text{ for } i \neq m \\ 0, & \text{otherwise.} \end{cases} \quad (37)$$

We now describe an efficient way to determine the vector elements (29) and matrix elements (30). Under the assumption that the policy $\pi_s(\mathbf{x})$ of the policy being evaluated are identical for each $\mathbf{x} \in \Omega^s(c)$, all coefficients of the polynomial expressions for $Q\mathbf{u}_h$, where $\mathbf{u}_h$ is a monomial or piecewise monomial basis vector, stays unchanged over each state set $\Omega^s(c)$. Denoting $\pi_s(c)$ the common policy in states $\mathbf{x} \in \Omega^s(c)$, we can express the elements of $Q\mathbf{u}_h$ in generic polyonmial form

$$[Q\mathbf{u}_h]_{\mathbf{x}} = \sum_{\nu \in E_{ch}} \zeta_{ch}(\nu) \prod_{i=1}^{G} {x'}_i^{\nu_i}, \quad \text{for } \mathbf{x} \in \Omega^s(c), c = 0, \ldots, \tilde{C}^s, \tag{38}$$

where $E_{ch} \subset \mathbb{N}^G, c = 0, \ldots, \tilde{C}^s$ are finite sets of exponent vectors, and $\zeta_{ch}(\nu)$ is the coefficient of the monomial $\prod_{i=1}^{G} {x'}_i^{\nu_i}$ within the set of states $\Omega^s(c)$.

By subsituting (38) we can rearrange (29) as [15]:

$$\begin{aligned}[U^T Q^T \mathbf{1}]_h &= \sum_{\mathbf{x} \in \Omega^s} [Q\mathbf{u}_h]_{\mathbf{x}} \\ &= \sum_{c=0}^{\tilde{C}^s} \sum_{\mathbf{x} \in \Omega^s} \sum_{\nu \in E_{ch}} \zeta_{ch}(\nu) \prod_{i=0}^{G} x_i^{\nu_i} \\ &= \sum_{c=0}^{\tilde{C}^s} \sum_{\nu \in E_{ch}} \zeta_{ch}(\nu) S(C^s, c, \nu)\end{aligned} \tag{39}$$

and (30) as

$$\begin{aligned}&[U^T Q^T Q U]_{wh} = \sum_{\mathbf{x} \in \Omega^s} [Q\mathbf{u}_w]_{\mathbf{x}} [Q\mathbf{u}_h]_{\mathbf{x}} \\ &= \sum_{c=0}^{\tilde{C}^s} \sum_{\mathbf{x} \in \Omega^s} \sum_{\nu \in E_{cw}} \zeta_{cw}(\nu) \prod_{i=0}^{G} x_i^{\nu_i} \sum_{\nu' \in E_{ch}} \zeta_{ch}(\nu') \prod_{i=0}^{G} x_i^{\nu_i'} \\ &= \sum_{c=0}^{\tilde{C}^s} \sum_{\nu \in E_{ch}} \sum_{\nu' \in E_{cw}} \zeta_{ch}(\nu) \zeta_{cw}(\nu') S(C^s, c, \nu + \nu')\end{aligned} \tag{40}$$

where $S(C^s, c, \nu)$ is defined as:

$$S(C^s, c, \nu) = \begin{cases} \sum\limits_{\mathbf{x} \in \Omega^s(c)} \prod\limits_{i=1}^{G} {x'}_i^{\nu_i}, & \text{C1} \\ \sum\limits_{\substack{\mathbf{x} \in \Omega^s(c) \\ x_n b_n \leq C^s}} \prod\limits_{i=1}^{G} {x'}_i^{\nu_i}, & \text{C2} \end{cases} \tag{41}$$

where C1 refers the condition $0 \leq c \leq C^s$ or $\nu = (\nu_1, \nu_2)$ with $\nu_2 \neq 0$ and C2 refers to the condition $C^s + 1 \leq c \leq \tilde{C}^s$ and $\nu = (\nu_1, 0)$ with $\nu_1 \neq 0$. A recursive procedure for computing the sum of monomials is formulated in the Appendix. With this procedure, the computational complexity for setting up the extended set of normal equations for link s becomes $O(G^4 \tilde{C}^s)$. Once the normal equations have been constructed, they can be solved by Cholesky factorization in O(H^3) operations. From the defintion of H in (28) this translates to O($G^6 + G^4 \tilde{C}^s + G^3 e^3$) operations.

VI. NUMERICAL RESULTS

A. Considered routing algorithms

The routing algorithms that are considered in the numerical evaluation are:

- MDP – MDP routing by reward maximization based on exact link model [11],
- MDP_P – MDP routing by cost minimization based on polynomial cost approximation described in Section V for loss-delay networks.
- MDP_D – MDP routing by reward maximization based on decomposition of the link Markov process proposed in [13].
- MDP_A – MDP routing based on Krishnan's and Hübner's state aggregation link model [8] with modified link reward parameters [12].
- LLR – Least Loaded Routing described in [3], [11].

The choice of basis vectors for the MDP_P routing algorithm correspond to method A in [15], see Table III. Method A performs least-squares approximation with a few basis vectors of all the considered types.

One simulation run with the MDP-based algorithms consists of an initial "warm up" period, followed by a number of adaptation periods, and finally a measurement period. Each adaptation period consists of an measurement period followed by a policy iteration step.

The performance of the Least Loaded Routing (LLR) method is also evaluated. The reason for evaluating LLR is that it is among the routing methods with best performance [1], [3]. The LLR routing method is implemented in many countries, including USA and Canada. We are not aware of any implementation of MDP routing in real networks.

B. Examples and results

The performance analysis is performed for the network example W6N described in Table I. The topology is fully connected. The total offered traffic load is measured by $\rho = \sum_{j \in J} b_j \lambda_j \mu_j^{-1}$ [Mbps*Erlang]. The link capacities and offered traffic volumes for network example W6N are based

	W6N
symmetrical	no
#nodes	6
#uni-directional links	30
#OD pairs P	30
#routes per OD pair	5
link capacity C^s [Mbps]	12-192
queue capacity L^s	0-3
network capacity [Mbps]	2484
max #links in path	2
#traffic categories G	2
mean holding time $1/\mu_j$ [s]	1, 10
bandwidth b_j [Mbps]	1, 6
total offered load ρ [Mbps*Erlang]	1816.8
$r'_j = r_j \mu_j / b_j$	1

TABLE I

DESCRIPTION OF NETWORK EXAMPLE W6N.

Link	Link capacity [Mbps]	Offered traffic [Mbps*Erlang]
1,2	36	32.96
1,3	24	8.36
1,4	162	154.68
1,5	48	24.56
1,6	48	34.93
2,3	96	30.13
2,4	96	121.93
2,5	108	92.14
2,6	96	99.07
3,4	12	14.30
3,5	48	8.23
3,6	24	15.90
4,5	192	95.30
4,6	84	99.60
5,6	168	76.27

TABLE II

LINK CAPACITY AND OFFERED TRAFFIC FOR W6N

on the example in [2] and is shown in Table II. Network W6N corresponds to a STM type network. The OD pairs in W6N are offered different traffic volumes (asymmetric case). The algorithm specific parameter settings, presented in Table III, were determined heuristically based on simulation experience.

We choose to evaluate all the routing algorithms in reward maximization framework, since this makes the comparison with our previous work easier. The routing performance is measured by the reward loss, average call set-up delay and objective reward loss:

$$L = 1 - \overline{R}/R, \tag{42}$$

$$\overline{D} = \sum_s \overline{D}_s \frac{\lambda_w^s}{\lambda_w}, \tag{43}$$

$$L_D = 1 - \overline{R}_D/R. \tag{44}$$

where $\overline{R}_D$ denotes the reward due to carried calls with penalty for delay:

$$\overline{R}_D = \overline{R} - \sum_s \alpha^s \overline{D}^s \frac{\lambda_w^s}{\lambda_w}. \tag{45}$$

and $\overline{R}$ denotes the reward due to carried calls:

MDP adaptation epochs	6
MDP_P adaptation epochs	4
MDP_D adaptation epochs	6
MDP_A adaptation epochs	4
call events in warm up period	500 000
call events in adaptation period	1000 000
call events in measurement period	1000 000
delay penalty weight α^s	100
#simulation points per curve N	4, 16, 19
#simulation runs per point M	20
Method A paramter D_1	2
Method A parameter P_1	1
Method A parameter D_2	1
Method A parameter E_2	1
Method A parameter e	$(1 + L^s)b_w$
Trunk res. (θ_n^n, θ_w^n), $\frac{\text{NB traffic}}{\text{WB traffic}} \leq 1$, $L^s = 0$	(6,0)
Trunk res. (θ_n^n, θ_w^n), otherwise	(0,0)

TABLE III

ALGORITHM SPECIFIC PARAMETERS

$$\overline{R} = \sum_{j \in J} r_j \overline{\lambda}_j, \quad (46)$$

where $\overline{\lambda}_j$ denotes the average class-j call acceptance rate.

C. Results Analysis

Fig. 1. Reward loss of different routing methods versus traffic ratio.

Fig. 2. Average call set-up delay of different routing methods versus traffic ratio.

Fig. 3. Objective reward loss of different routing methods versus traffic ratio.

VII. Conclusion

In this paper we formulated the CAC and routing problem in the cost minimization MDP framework with costs incurred by call blocking and call set up delay. We extended Rummukainen's and Virtamo's polynomial cost approximation framework to mixed loss-delay networks. The contribution includes a link model for loss-delay call set up, and a numerical procedure for setting up the equations in this link model.

Appendix

Computing sums of monomials

In this Appendix we show that $S(a,c,\nu)$ can be expressed recursively as

$$S(a,c,\nu) = \begin{cases} \sum_{\theta=0}^{\nu_k} \binom{\nu_k}{\theta} S(a, c-b_k, \nu+(\theta-\nu_k)\mathbf{e}_k) & \text{C1} \\ \sum_{\theta=0}^{\nu_1} \binom{\nu_1}{\theta} \tilde{S}(a-b_n, c-b_n, \theta\mathbf{e}_1) & \text{C2} \end{cases} \qquad (47)$$

where

$$\tilde{S}(a,c,\nu) = \sum_{\theta=0}^{\nu_1} \binom{\nu_1}{\theta} \tilde{S}(a-b_n, c-b_n, \theta\mathbf{e}_1) \qquad (48)$$

Let m be an index such that $\nu_m > 0$; there is clearly one since $\nu \neq \mathbf{0}$. By reducing the value of x'_m we can write $S(a,c,\nu)$ when condition C1 is fulfilled as

$$\begin{aligned} S(a,c,\nu) &= \sum_{\mathbf{x}^T\mathbf{b}=c} \prod_{i=1}^{G} {x'}_i^{\nu_i} \\ &= \sum_{\mathbf{x}^T\mathbf{b}=c-b_m} (x'_m+1)^{\nu_i} \prod_{\substack{i=1 \\ i\neq m}}^{G} {x'}_i^{\nu_i} \\ &= \sum_{\mathbf{x}^T\mathbf{b}=c-b_m} \left(\sum_{\theta=0}^{\nu_m} \binom{\nu_m}{\theta} {x'}_m^{\theta} \right) \prod_{\substack{i=1 \\ i\neq m}}^{G} {x'}_i^{\nu_i} \\ &= \sum_{\theta=0}^{\nu_m} \binom{\nu_m}{\theta} S(a, c-b_m, \nu+(\theta-\nu_m)\mathbf{e}_m) \end{aligned} \qquad (49)$$

Similarly, we write for condition C2

$$\begin{aligned} S(a,c,\nu) &= \sum_{\substack{\mathbf{x}^T\mathbf{b}=c \\ x_n b_n \le a}} {x'}_n^{\nu_1} = \sum_{\substack{\mathbf{x}^T\mathbf{b}=c-b_n \\ x_n b_n \le a-b_n}} (x'_1+1)^{\nu_1} \\ &= \sum_{\theta=0}^{\nu_1} \binom{\nu_1}{\theta} \tilde{S}(a-b_n, c-b_n, \theta\mathbf{e}_1) \end{aligned} \qquad (50)$$

The ground cases for the recursions are

$$\begin{aligned} S(a,c,\nu) &= 0, && c<0, \\ S(a,c,\mathbf{0}) &= s(c,w), && 0 \le c \le \tilde{C}^s, \\ \tilde{S}(a,c,\nu) &= 0, && c \le \max(0,a), \\ \tilde{S}(a,c,\mathbf{0}) &= \tilde{s}(a,c,w), && 0 \le a \le c \le \tilde{C}^s, \end{aligned} \qquad (51)$$

where $s(c,w)$ is defined by the recursion

$$\begin{aligned} s(c,w) = & \sum_{\mathbf{x}^T\mathbf{b}=c} 1 = \sum_{\substack{\mathbf{x}^T\mathbf{b}=c \\ x_w=0}} 1 + \sum_{\substack{\mathbf{x}^T\mathbf{b}=c \\ x_w>0}} 1 \\ & = \sum_{x_n b_n = c} 1 + \sum_{\mathbf{y}^T\mathbf{b}=c-b_w} 1 \\ & = s(c,n) + s(c-b_w, w) \end{aligned} \qquad (52)$$

and $\tilde{s}(a,c,w)$ is defined by the recursion

$$\begin{aligned} \tilde{s}(a,c,w) = & \sum_{\substack{x_n b_n = c \\ x_n b_n \le a}} 1 + \sum_{\substack{\mathbf{y}^T\mathbf{b}=c-b_w \\ y_n b_n \le a}} 1 \\ & = \tilde{s}(a,c,n) + \tilde{s}(a, c-b_w, w). \end{aligned} \qquad (53)$$

The ground cases for $s(c,i)$ are:

$$\begin{aligned} s(0,i) &= 1 && \text{for } i \in I, \\ s(c,i) &= 0 && \text{for } c<0, i \in I, \\ s(c,1) &= 1_{b_n|c} && \text{for } c>0. \end{aligned} \qquad (54)$$

The ground cases for $\tilde{s}(a,c,i)$ are:

$$\begin{aligned} \tilde{s}(a,0,i) &= 1 && \text{for } i \in I, \\ \tilde{s}(a,c,i) &= 0 && \text{for } c<0, i \in I, \\ \tilde{s}(a,c,1) &= 1_{b_n|c} && \text{for } 0 < c \le a. \end{aligned} \qquad (55)$$

Acknowledgement

We acknowledge the advice from Hannu Rummukainen.

References

[1] Ash G. *Dynamic Routing in Telecommunications Networks*, McGraw-Hill, New York, 1998.

[2] Chung S., Kashper A., Ross K. Computing Approximate Blocking Probabilities for Large Loss Networks with State-Dependent Routing, *IEEE/ACM Transactions on Networking*, 1993; 1(1):105-115.

[3] Dziong Z. *ATM Network Resource Management*. McGraw-Hill (ISBN 0-07-018546-8), 1997.

[4] Dziong Z., Liao K-Q., Mason L.G. Flow Control Models for Multi-Service Networks with Delayed Call Set Up, In *Proceedings of IEEE INFOCOM'90*, IEEE Computer Society Press, 1990; 39-46.

[5] Dziong Z., Mason L. Call admission and routing in multi-service loss networks, *IEEE Transactions on Communications*, 1994; **42**(2):2011-2022.

[6] Howard R. A. *Dynamic Programming and Markov Process*, The M.I.T. Press, Cambridge, Massachusetts, 1960.

[7] Kaufman J. Blocking in a shared resource environment, *IEEE Transactions on Communications*, 1981; **COM-29**(10): 1474-1481.

[8] Krishnan K., Hübner F. Admission control and routing for multirate circuit-switched traffic, In *Proceedings of ITC'15*, Washington, USA, 1997.

[9] Liao K.-Q., Mason L.G. An Approximate Performance Model for a Multislot Integrated Services System, *IEEE Transactions on Communications*, 1989; 37 (3): 211- 221.

[10] Marbach P., O. Mihatsch, Tsitsiklis J. Call admission control and routing in integrated services networks using neuro-dynamic programming, *IEEE Transactions on Communications*, 2000; **18**(2): 197-208.

[11] Nordström E., Dziong Z. CAC and Routing in Multi-Service Networks with Blocked Wide-Band Calls Delayed, Part I: Exact Link MDP Framework, Accepted for publication in *European Transactions on Telecommunications*, 2005.

[12] Nordström E., Carlström J. A New Reward Model for MDP State Aggregation with Application to CAC and Routing, Accepted for publication in *European Transactions on Telecommunications*, 2005.

[13] Nordström E., Dziong Z. CAC and Routing in Multi-Service Networks with Blocked Wide-Band Calls Delayed, Part II: Approximate Link MDP Framework, Accepted for publication in *European Transactions on Telecommunications*, 2005.

[14] Roberts J. A service system with heterogeneous user requirements, *Performance of Data Communications Systems and Their Applications*, Editor G. Pujolle, North-Holland, 1981.

[15] Rummukainen H., Virtamo J. Polynomial cost approximations in Markov decision theory based call admission control, *IEEE/ACM Transactions on Networking*, 2001; **9**(6):769-779.

[16] De Serres Y., Mason L.G. A multi-server queue, with narrow- and wide-band customers and WB restricted access, *IEEE Transactions on Communications*, 1988; 36(6):675-684.

[17] Tijms H. *Stochastic Modeling and Analysis – a Computational Approch*, Wiley, 1986.

PLACE
PHOTO
HERE

Ernst Nordström received his M.Sc. degree in Engineering Physics in 1992, and his Ph.D. degree in Computer Systems in 1998, both from Uppsala University, Sweden. In 1996 and 2000 he was on sabbatical leaves at the Department of Telecommunication, Technical University of Denmark, Lyngby, Denmark, and at the Ericsson Traffic Lab, Budapest, Hungary. In 2000 he joined the Culture, Media and Computer Science Department at Dalarna University, Sweden as a senior lecturer. His research interests include Markov decision process theory and optimization theory with application to CAC, routing, and dimensioning in multi-service networks.