EUROPEAN TRANSACTIONS ON TELECOMMUNICATIONS
Euro. Trans. Telecomms. 2005; **16**:495–508
Published online 29 September 2004 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/ett.1007

Communication Networks

A new reward model for MDP state aggregation with application to CAC and Routing

Ernst Nordström[1]* and Jakob Carlström[2]

[1] *Department of Culture/Media/Computer Science, Dalarna University, SE-781 88 Borlänge, Sweden*
[2] *Xelerated, Olof Palmes gata 29, SE-111 22 Stockholm, Sweden*

SUMMARY

An optimal solution of the call admission control and routing problem in multi-service loss networks, in terms of average reward per time unit, is possible by modeling the network behavior as a Markov decision process (MDP). However, even after applying the standard link independence assumption, the solution of the corresponding set of link problems may involve considerable numerical computation. In this paper, we study an approximate MDP framework on the link level, where vector-valued MDP states are mapped into a set of aggregate scalar MDP states corresponding to link occupancies. In particular, we propose a new model of the expected reward for admitting a call on the network. Compared to Krishnan's and Hübner's method [11], our reward model more accurately reflects the bandwidth occupancy by different call categories. The exact and approximate link MDP frameworks are compared by simulations, and the results show that the proposed link reward model significantly improves the performance of Krishnan's and Hübner's method. Copyright © 2004 AEIT.

1. INTRODUCTION

We study the problem of optimal call admission control (CAC) and routing in multi-service loss 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. First, CAC determines the set of feasible paths between the source and destination which offer sufficient QoS to the new and existing calls in terms of delay, jitter and data loss. Second, the network should choose to reject the call or to accept it on some path among the set of feasible paths. 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.

Modern CAC and routing mechanisms are state-dependent rather than static, which means 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. A state-dependent CAC and routing policy is a mapping, for every call class, from a network state space to a set of possible routing decisions, see Figure 1. A state-dependent mechanism 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)

* Correspondence to: Ernst Nordström, Department of Culture/Media/Computer Science, Dalarna University, SE-781 88 Borlänge, Sweden. E-mail: eno@du.se

Contract/grant sponsors: Swedish Knowledge Foundation; Ericsson Telecom AB.

Received 11 August 2003
Revised 1 December 2003
Accepted 18 February 2004

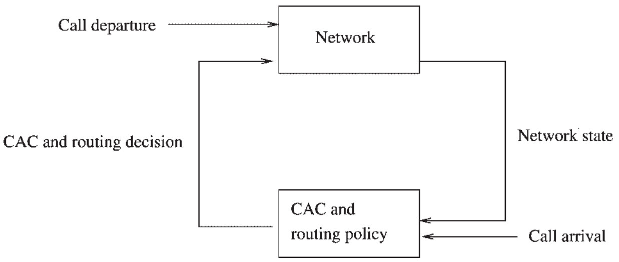

Figure 1. State-dependent call admission control (CAC) and routing.

[4, 19]. 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. In the MDP framework, each call is described by a expected reward parameter and the objective is to maximize the reward from carried calls.

The expected reward parameter is a versatile tool for controlling the operator revenue and GoS. The reward parameter can be set proportional to the user charge of a call with mean holding time. Alternatively, the reward parameter can be used to control the distribution of call blocking probabilities among the call classes.

Optimal state-dependent CAC and routing policies can be computed using an exact network MDP framework. However, the cardinality of the network state and policy spaces in the exact framework can be very large even for moderate-size networks, making the computational cost of MDP-based CAC and routing prohibitive. The objective of the work presented in this paper is to reduce the computational cost to manageable levels, while retaining the performance of MDP-based CAC and routing.

A necessary modeling simplification is to decompose the network into a set of links assumed to have independent traffic and reward processes respectively. When formulating the MDP framework for each link, calls with the same bandwidth requirement are aggregated into a common category, which corresponds to one dimension in the link state vector.

The computational burden of each link MDP task increases when the number of categories increases, or when the ratio between the link capacity and the bandwidth requirement increases for some of the call categories. In the first case, the increase is exponential, in the second case it is polynomial. Simplified link MDP frameworks with reduced computational cost have been proposed, including methods based on state aggregation [9, 11], decomposition of the link Markov process [16] and polynomial cost approximation [13, 18].

Hwang, Kurose and Towsley proposed a simplified link MDP framework based on state aggregation with scalar link state representing the link occupancy [9]. A birth–death process based on Pascal approximation [3] drives the one-dimensional Markov chain. The MDP task is solved by a one-step policy iteration algorithm.

Also Krishnan and Hübner proposed a simplified link MDP framework based on state aggregation with scalar link state representing the link occupancy [11]. Transition probabilities between link states were derived from link occupancy probabilities obtained by a recursive procedure due to Kaufman [10] and Roberts [17]. The MDP task was also solved by one-step policy iteration.

Dziong, Liao and Mason proposed a simplified link MDP framework based on decomposition of the link Markov process into per-category link Markov processes [5]. They observed that if the holding times of wide-band (WB) calls are significantly longer than for narrow-band (NB) calls, the NB process changes state much more often than the WB process. This suggests 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.

The CAC and routing problem for networks with blocked NB calls cleared and blocked WB calls delayed was studied by Nordström and Dziong [15, 16]. In this case, stage aggregation cannot be used due to the fact that the state space is not coordinate convex. According to simulation results in Reference [16], the decomposition method is relatively efficient in case of delayed WB call set up, but can fail considerably in case of pure loss networks.

Marbach, Mihatsch and Tsitsiklis applied reinforcement learning to estimate the optimal second-degree polynomial link-cost approximation [13]. 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 framework for computing the cost relative values as a linear combination of a modest number of basis vectors [18]. Single-coordinate and double-coordinate monomial vectors up to some degree, were considered as basis vectors. The computational complexity of determining the coefficients of the polynomials was identical to the complexity of Krishnan's and Hübner's method, i.e. proportional to the capacity of the link.

The contribution of this paper is threefold. First, we propose a new model of the link reward to be used with the state aggregation method by Krishnan and Hübner [11]. Second, we formulate Krishnan's and Hübner's link MDP framework in terms of reward maximization rather than in terms of cost minimization. Third, we present an extensive numerical evaluation, based on simulation, of a set of MDP-based routing algorithms and a conventional routing method called least loaded routing (LLR).

The numerical results show that the modified link reward model significantly improves the average reward rate, compared to the original model used by Krishnan and Hübner. Moreover, Krishnan's and Hübner's state aggregation method, with our modified link reward model, yields higher reward rate than Hwang's, Kurose's and Towsley's state aggregation method. The best MDP methods outperform the conventional routing method (LLR).

A numerical comparison to the polynomial cost approximation by Rummukainen and Virtamo is left for future work.

The paper is organized as follows: Section 2 formulates the CAC and routing problem in terms of offered traffic, network model and optimization objective. Section 3 describes the network, exact link MDP model and approximate link MDP models based on state aggregation. Section 4 outlines the MDP computation procedure. Section 5 presents the numerical results of the performance of MDP-based routing as well as LLR routing. Finally, Section 6 concludes the paper.

2. PROBLEM FORMULATION

The network is assumed to consist of a set of switching nodes, interconnected by bi-directional links according to some network topology. Each bi-directional link consists of two uni-directional links, carrying traffic in opposite directions.

The network is offered traffic from K classes which are, for sake of simplicity, assumed to be subject to deterministic multiplexing. The jth class is characterized by the following:

- origin–destination (OD) node pair,
- bandwidth requirement b_j [Mbps],
- Poissonian call arrival process with rate λ_j [s^{-1}],
- exponentially distributed call holding time with mean $1/\mu_j$ [s],
- set of alternative routes, W_j and
- reward parameter $r_j \in (0, \infty)$

The classes are classified into G bandwidth categories. The ith category is characterized by:

- bandwidth requirement b_i [Mbps],
- average mean call holding time $1/\overline{\mu}_i$ [s] and
- average reward parameter $\overline{r}_i$.

The task is to find an optimal routing policy π^* which maximizes the mean reward from the network, defined as:

$$\overline{R}(\pi) = \sum_{j \in J} r_j \overline{\lambda}_j \tag{1}$$

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

3. MDP MODELING

3.1. Network decomposition

In the exact MDP framework, the network state and policy 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 [7].

The network Markov process is decomposed into a set of independent link Markov processes, driven by state-dependent Poisson call arrival processes with rate $\lambda_j^s(\mathbf{x}, \pi)$, where s denotes the link index, $\mathbf{x}$ denotes the link state and π denotes the CAC and routing policy. 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

$$r_j = \sum_{s \in S_k} r_j^s(\pi) \tag{2}$$

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 [7]. 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 of links in the call's path.

Even in the decomposed model, the state space can be quite large if 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

$i \in I = \{1, \ldots, G\}$ with average reward parameter defined as [7]:

$$\overline{r}_i^s(\pi) = \frac{\sum_{j\in J_i} r_j^s(\pi)\overline{\lambda_j^s}(\pi)}{\sum_{j\in J_i}\overline{\lambda_j^s}(\pi)} \tag{3}$$

where J_i denotes the set of classes that belongs to the ith 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.

3.2. Exact link MDP model

This section describes the exact link MDP model, which provides the basis for the MDP computational procedure presented in the next section. The state in the exact link model is given by $\mathbf{x} = \{x_i\}$, where x_i denotes the number of category i calls on the link. The state space X for the exact link model is given by:

$$X = \left\{\mathbf{x} = \{x_i\} : \sum_{i\in I} b_i x_i \leqslant C^s\right\} \tag{4}$$

where C^s denotes the capacity of link s.

It can be shown that the size of the state space grows like:

$$S \sim \frac{1}{G!}\prod_{i\in I}(N_i^s + 1) \tag{5}$$

where $N_i^s = \lfloor C^s/b_i \rfloor$ denotes the maximal number of category i calls on the link.

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

$$A = \{a = \{a_i\} : a_i \in \{0, 1\}, i \in I\} \tag{6}$$

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:

$$A(\mathbf{x}) = \{a \in A : a_i = 0 \text{ if } \mathbf{x} + \delta_i \notin X, i \in I\} \tag{7}$$

where δ_i denotes a vector of zeros except a one in position $i \in I$.

The Markov chain is characterized by state transition probabilities $p_{xy}(a)$ which express the probability that the next state is $\mathbf{y}$, given that action a is taken in state $\mathbf{x}$. In our case, the state transition probabilities become:

$$p_{xy}(a) = \begin{cases} \lambda_i^s(\mathbf{x},\pi)a_i\tau(\mathbf{x},a), & \mathbf{y} = \mathbf{x} + \delta_i \in X, i \in I \\ x_i\overline{\mu}_i\tau(\mathbf{x},a), & \mathbf{y} = \mathbf{x} - \delta_i \in X, i \in I \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where $\lambda_i^s(\mathbf{x},\pi)$ denotes the ith category arrival rate to the link in state $\mathbf{x}$ under routing policy π, $\overline{\mu}_i$ denotes the average departure rate of category i calls, and $\tau(\mathbf{x},a)$ denotes the average sojourn time in state $\mathbf{x}$. The link call arrival rates, $\lambda_i^s(\mathbf{x},\pi)$, are given by:

$$\lambda_i^s(\mathbf{x},\pi) = \sum_{j\in J_i}\lambda_j^k(\pi)\phi_j^s(\mathbf{x},\pi)\prod_{c\in S_k\setminus\{s\}}(1 - B_j^c(\pi)) \tag{9}$$

where $s \in S_k$, $B_j^c(\pi)$ denotes the probability that link c has not enough capacity to accept a class j call, and $\phi_j^s(\mathbf{x},\pi)$ denotes a filtering probability defined as:

$$\phi_j^s(\mathbf{x},\pi) = P\left\{\sum_{c\in S_k\setminus\{s\}} p_j^c(\mathbf{x},\pi) < r_j - p_j^s(\mathbf{x},\pi) \mid \overline{B}_j\right\} \tag{10}$$

where $\overline{B}_j$ denotes the condition that no link on path k is in the blocking state (note that $p_j^s(\mathbf{x},\pi)$ is constant in Equation (10)). In other words, $\phi_j^s(\mathbf{x},\pi)$ is the probability that the path net-gain is positive (on condition that there is enough path capacity to carry the call). The filtering probability can be computed using link state distributions [8], or approximated with one according to experiments in Reference [7]. The $\lambda_j^k(\pi)$ denote the arrival rate of class j to path $k \in W_j$, and is given by the following load sharing model [7]:

$$\lambda_j^k(\pi) = \lambda_j\frac{\overline{\lambda}_j^k(\pi)}{\sum_{h\in W_j}\overline{\lambda}_j^h(\pi)} \tag{11}$$

where $\overline{\lambda}_j^k(\pi)$ denotes the average rate of accepted class j calls on path k, and λ_j denotes the arrival rate of class j.

The average departure rate for the ith category is computed as:

$$\overline{\mu}_i = \left[\sum_{j\in J_i} p_{ij}\mu_j^{-1}\right]^{-1} \tag{12}$$

where p_{ij} denotes the probability that an arbitrary ith category call found on the link is from class $j \in J_i$:

$$p_{ij} = \frac{\overline{\lambda}_j^s(\pi)}{\sum_{c\in J_i}\overline{\lambda}_c^s(\pi)} \tag{13}$$

where $\overline{\lambda}_j^s(\pi)$ denotes the average class j call acceptance rate on link s.

The average sojourn time $\tau(\mathbf{x},a)$ in state $\mathbf{x}$ is given by:

$$\tau(\mathbf{x},a) = \left\{\sum_{i\in I} x_i\overline{\mu}_i + a_i\lambda_i^s(\mathbf{x},\pi)\right\}^{-1} \tag{14}$$

The expected reward in state $\mathbf{x}$ is given by $R^s(\mathbf{x}, a) = \rho^s(\mathbf{x})\tau(\mathbf{x}, a)$, where $\rho^s(\mathbf{x})$ is obtained from

$$\rho^s(\mathbf{x}) = \sum_{i \in I} \overline{r}_i^s x_i \overline{\mu}_i \tag{15}$$

3.3. New approximate link MDP model

In the aggregate state link model, the G-dimensional micro-state is aggregated into a one-dimensional macro-state m [11]:

$$m = \sum_{i \in I} b_i x_i \tag{16}$$

The macro-state process $\{m\}$ does not, in general, form a Markov process, since future evolution form state m will clearly depend on the sample path history (characterized by the micro-states $\mathbf{x}$) into state m. The set of possible states is denoted M:

$$M = \left\{ m \,:\, m = \sum_{i \in I} b_i x_i \leqslant C^s, x_i \geqslant 0 \right\} \tag{17}$$

The size of the state space is significantly smaller than for the exact link model. For example, if the bandwidth requirements are integer valued, and at least one of the categories requires one bandwidth unit, the state space will contain $S = C^s + 1$ states.

The set of permissible actions in state m is denoted $A\,(m)$:

$$A\,(m) = \{a \in A \,:\, a_i = 0 \text{ if } m + b_i \notin M, i \in I\} \tag{18}$$

We approximate the macro-state process by a Markov process with the following state transition probabilities:

$$p_{mn}(a) = \begin{cases} \lambda_i^s(m, \pi) a_i \tau(m, \pi), & n = m + b_i \in M, i \in I \\ E[x_i \,|\, m] \overline{\mu}_i \tau(m, \pi), & n = m - b_i \in M, i \in I \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

where $\lambda_i^s(m, \pi)$ is given by a formula analogous to Equation (9), and $E[x_i \,|\, m]$ is the expected number of category i calls when the link occupancy is m. According to Reference [10], $E[x_i \,|\, m]$ can be approximated by

$$\lambda_i(m - b_i, \pi) q(m - b_i) = \overline{\mu}_i E[x_i \,|\, m] q(m) \tag{20}$$

which is exact for the complete sharing link access policy, i.e. when calls always are accepted if there is sufficient free capacity. The link occupancy equilibrium distribution, $\{q(m)\}$, is given by [10, 17]:

$$m q\,(m) = \sum_{i \in I} \frac{\lambda_i^s(m - b_i, \pi)}{\overline{\mu}_i} b_i q(m - b_i) \tag{21}$$

where $q\,(m) = 0$ for all $m < 0$. The recursion starts with $q\,(0) = 1$ and ends with normalizing the results for obtaining the probabilities. The recursion is exact for the complete sharing access policy.

The expected sojourn time in state m is:

$$\tau(m, a) = \left\{ \sum_{i \in I} E[x_i \,|\, m] \overline{\mu}_i + a_i \lambda_i^s(m, \pi) \right\} \tag{22}$$

The expected reward delivered when leaving state m is given by $R^s(m, a) = \rho^s(m)\tau(m, a)$, where the reward accumulation rate $\rho^s(m)$ is given by:

$$\rho^s(m) = \sum_{i \in I} \tilde{r}_i^s(\pi) E[x_i \,|\, m] \overline{\mu}_i \tag{23}$$

where $\tilde{r}_i^s(\pi)$ is the modified link reward parameter for category i. Below, we formally explain the formula for the reward accumulation rate we make the following remarks. In the state aggregation MDP model, the basic modeling entity is the link occupancy m, not the number of calls from each category x_i. The bandwidth m is shared, in a statistical sense, between all call categories. We can only express the probability $P[i \,|\, m]$ that a unit of bandwidth is occupied by category i, given that the link occupancy is m. Note that in the exact link model, this statistical sharing does not occur—the bandwidth b_i of a category i is only used by this category.

The reward accumulation rate is obtained as a sum of average per-category reward rates, $\tilde{r}_i^s(\pi) E[x_i \,|\, m] \overline{\mu}_i$. Here, $\tilde{r}_i^s(\pi)$ denotes the reward obtained when serving a category i call in the state aggregation model. Due to the statistical sharing of bandwidth b_i among the G different categories on the link, we make the important observation that every category $c \in I$ will contribute to the average reward $\tilde{r}_i^s(\pi)$. We call the quantity $\tilde{r}_i^s(\pi)$ the modified reward, to distinguish it from the original reward $\overline{r}_i^s(\pi)$ obtained in Equation (3) which offers dedicated portions of bandwidths to each category. Formally, the modified reward parameter, is defined as:

$$\tilde{r}_i^s(\pi) = \tilde{r}_i^s(\pi, b_i) = \sum_{m \in M} \sum_{c \in I} \overline{r}_c^s(\pi) E[z_c \,|\, m, b_i] q(m) \tag{24}$$

where $E[z_c \,|\, m, b_i]$ denotes the expected number of category c calls which occupy a bandwidth portion b_i [Mbps], given that the link occupancy is m.

The expectation $E[z_c \,|\, m, b_i]$ is determined from the probability $P[c \,|\, m]$ of finding a unit bandwidth occupied by category c, given that the link occupancy is m, multiplied by the number of category c calls that fits into the bandwidth portion of a category i call:

$$\tilde{r}_i^s(\pi) = \sum_{m \in M} \sum_{c \in I} \overline{r}_c^s(\pi) P[c \mid m] \frac{b_i}{b_c} q(m) = \sum_{c \in I} \overline{r}_c^s(\pi) P(c) \frac{b_i}{b_c} \tag{25}$$

Finally, we use the fact that the probability $P(c)$ of finding a category c call on the link is proportional to the relative portion of accepted category c calls:

$$\tilde{r}_i^s(\pi) = \sum_{c \in I} \overline{r}_c^s(\pi) \frac{\overline{\lambda}_c^s(\pi)}{\sum_{d \in I} \overline{\lambda}_d^s(\pi)} \frac{b_i}{b_c} \tag{26}$$

Note that the fraction of the bandwidth b_i [Mbps] which is devoted, on average, to category c will change every time the link occupancy m changes. Our proposed reward model gives the average contribution of category c over the whole state space M. Formally, the average contribution is computed as weighted sum over the state space, with weights given by the probabilities $q(m)$.

Figure 2 illustrates the computation of macro-scale bandwidth shares from micro-scale bandwidth shares, in an example where $m = 24$ is shared equally among two categories.

The fact that state aggregation employs one-step policy iteration 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.

3.4. Approximate link MDP model by Hwang, Kurose and Towsley

In Reference [9], Hwang, Kurose and Towsley proposed a link MDP framework with scalar state representing the link occupancy. A birth-death process based on the Pascal approximation [3] drives the one-dimensional Markov chain. Without loss of generality, the authors assume that the bandwidth requirements of the call categories are ordered as $1 \leqslant b_1 \leqslant b_2 \leqslant \cdots \leqslant b_{G-1} \leqslant b_G$ and the holding time of the first category, $1/\overline{\mu}_1$, is normalized to 1.

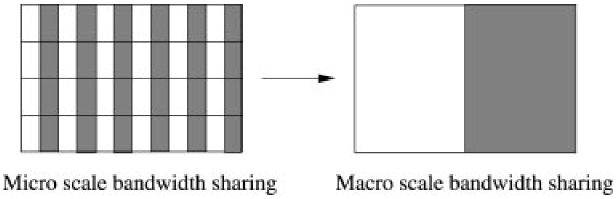

Figure 2. The aggregate bandwidth m on the macro-bandwidth scale is shared between the categories i in proportion to the probability $P[i \mid m]$ of occupying a single bandwidth unit on the micro-bandwidth scale.

Transitions to the next state above the current state are always allowed, and transitions to the state below the current state occur at unit rate. The state transition probabilities are as follows:

$$p_{mn} = \begin{cases} \Lambda^s(m,\pi)\tau(m), & n = m+1 \in M, i \in I \\ m\tau(m), & n = m-1 \in M, i \in I \\ 0, & \text{otherwise} \end{cases} \tag{27}$$

where the birth rate $\Lambda^s(m,\pi)$ is given by:

$$\Lambda^s(m,\pi) = \frac{\xi^s(m,\pi)^2}{\sigma^s(m,\pi)^2} + m\left[1 - \frac{\xi^s(m,\pi)}{\sigma^s(m,\pi)^2}\right] \tag{28}$$

where the quantities $\xi^s(m,\pi)$ and $\sigma^s(m,\pi)$ are given by:

$$\xi^s(m,\pi) = \sum_{i \in I} \frac{b_i \lambda_i^s(m,\pi)}{\overline{\mu}_i} \tag{29}$$

$$\sigma^s(m,\pi)^2 = \sum_{i \in I} \frac{b_i^2 \lambda_i^s(m,\pi)}{\overline{\mu}_i} \tag{30}$$

where $\lambda_i^s(m,\pi)$ is given by a formula analogous to Equation (9). The expected sojourn time in state m is given by:

$$\tau(m) = \{m + \Lambda^s(m,\pi)\}^{-1} \tag{31}$$

The MDP framework proposed by Hwang, Kurose and Towsley [9] is based on cost minimization instead of reward maximization. The expected cost incurred in state m is given by $W^s(m) = \rho^s(m)\tau(m)$, where $\rho^s(m)$ denotes the cost accumulation rate. Hwang, Kurose and Towsley originally define the cost accumulation rate as follows [9]:

$$\rho^s(m) = \begin{cases} 0, & 0 \leqslant m \leqslant C^s - b_G \\ \sum_{i \in I} \alpha_i \eta_i^s(m,\pi), & C^s - b_i < m \leqslant C^s - b_{i-1} \\ \sum_{i \in I} (1-\alpha_i)\eta_i^s(m,\pi), & m = C^s \end{cases} \tag{32}$$

where $\eta_i^s(m,\pi) = \overline{r}_i^s(\pi)\lambda_i^s(m,\pi)$ and α_i is a category-dependent heuristic parameter which always is 0 for $i = 1$ and $\alpha_i > 0$ for $i > 1$.

3.5. Approximate link MDP model by Krishnan and Hübner

Krishnan's and Hübner's MDP framework [11] is also based on state aggregation and cost minimization. The state space and the state transition probabilities are the same as for the MDP framework presented in Section 3.3, which was based on reward maximization.

The expected cost incurred in state m is given by $W^s(m,a) = \rho^s(m,a)\tau(m,a)$, where the cost accumulation rate $\rho^s(m,a)$ is given by:

$$\rho^s(m,a) = \sum_{i\in I} \lambda_i^s(m,\pi)\bar{r}_i^s(\pi)(1-a_i) \tag{33}$$

where $a = \{a_i\} = \pi_s(m)$ denotes the link CAC decision in state m. Thus, this cost formulation does not model the statistical resource sharing discussed in Section 3.3. Apart from this and the employment of cost minimization, the frameworks are identical.

4. MDP COMPUTATIONAL PROCEDURE

This section outlines the MDP computational procedure for determining a near-optimal CAC and routing policy using the exact link model. The following formulas are also valid for the link models based on state aggregation with the modified reward parameter provided the following variable substitutions are done:

- decomposed network state: $\mathbf{y} \to \mathbf{n}$,
- link state: $\mathbf{x} \to m$,
- reward parameter: $\bar{r}_i^s(\pi) \to \tilde{r}_i^s(\pi)$ and
- increment in link state: $\delta_i \to b_i$.

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 (see Figure 3). The call is rejected if the path net-gain is negative, or if no path would offer sufficient QoS.

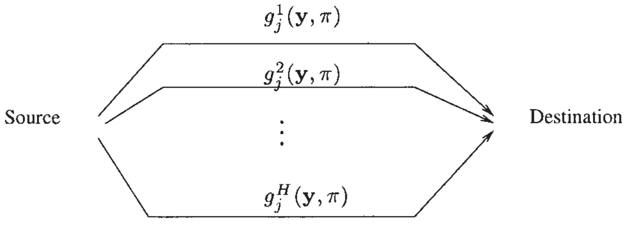

Figure 3. The call is offered to a path which has sufficient QoS and maximal positive path net-gain among the $H = |W_j|$ alternative paths.

4.1. Basic definitions

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

$$g_j^k(\mathbf{y},\pi) = r_j - \sum_{s\in S_k} p_i^s(\mathbf{x},\pi) \tag{34}$$

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 ith category call in state $\mathbf{x} = \{x_i\}$. In the reward maximization MDP framework the link shadow price is defined as follows:

$$p_i^s(\mathbf{x},\pi) = \bar{r}_i^s(\pi) - g_i^s(\mathbf{x},\pi) \tag{35}$$

where $g_i^s(\mathbf{x},\pi)$ denotes the link net-gain for admission of a category i call in state $\mathbf{x}$. The link net-gain expresses the increase in long-term reward due to admission of a category i call in link state $\mathbf{x}$ and is defined, for the reward maximization MDP framework, as follows:

$$g_i^s(\mathbf{x},\pi) = v^s(\mathbf{x}+\delta_i,\pi) - v^s(\mathbf{x},\pi) \tag{36}$$

where $v^s(\mathbf{x},\pi)$ denotes the relative value for category i in state $\mathbf{x}$ and δ_i denotes a vector of zeros except for a one in position i.

In the cost minimization, MDP framework the link shadow price is defined as follows:

$$p_i^s(\mathbf{x},\pi) = v^s(\mathbf{x}+\delta_i,\pi) - v^s(\mathbf{x},\pi) \tag{37}$$

To give more insight into the definition of relative values, let us define the expected link reward, $R^s(x_0,\pi,T)$, obtained in an interval (t_0, t_0+T) of length T, assuming state x_0 at time t_0:

$$R^s(\mathbf{x}_0,\pi,T) = E\left[\int_{t_0}^{t_0+T} q^s(\mathbf{x}(t))\,\mathrm{d}t\right] \tag{38}$$

where $q^s(\mathbf{x}(t))$ denotes the reward accumulation rate in state $\mathbf{x}(t)$. The process $\{\mathbf{x}(t)\}$ is driven by a probabilistic law of motion specified by certain state transition probabilities.

The relative value can now be written as:

$$v^s(\mathbf{x}_0,\pi) = \lim_{T\to\infty}[R^s(\mathbf{x}_0,\pi,T) - R^s(\mathbf{x}_r,\pi,T)] \tag{39}$$

That is, the relative value in state $\mathbf{x}_0$ is defined as the difference in future reward earnings 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 below).

4.2. 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 $B_j^c(\pi)$ 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 reward rate $\overline{R}^s(\pi)$ 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 reward rate.
4. *Convergence test*: Repeat from 2 until average reward 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 [19].

4.2.1. Value determination. The value determination step for link s determines the relative values $v^s(\mathbf{x},\pi)$ for all states $\mathbf{x}\in X$ by solving a sparse system of linear equations:

$$\begin{cases} v^s(\mathbf{x},\pi)=R^s(\mathbf{x},a)-\overline{R}^s(\pi)\tau(\mathbf{x},a)+\sum_{\mathbf{y}\in X}p_{xy}(a)v^s(\mathbf{y},\pi) \\ v^s(\mathbf{x}_r,\pi)=0; \qquad \mathbf{x}\in X \end{cases} \tag{40}$$

where the following quantities need to be specified:

- X: the state space, i.e. the set of possible states,
- $a=\pi_s(\mathbf{x})$: the control action in state $\mathbf{x}$,
- $\tau(\mathbf{x},a)$: the expected sojourn time in state $\mathbf{x}$,
- $R^s(\mathbf{x},a)$: the expected link reward when leaving state $\mathbf{x}$,
- $p_{xy}(a)$: the transition probability from state $\mathbf{x}$ to state $\mathbf{y}$, given that action a is taken in state $\mathbf{x}$,
- $\mathbf{x}_r$: the reference state (e.g. the empty state),

in order to compute the unknowns:

- $v^s(\mathbf{x},\pi)$: the relative value in state $\mathbf{x}$ under routing policy π,
- $\overline{R}^s(\pi)$: the average rate of link reward under policy π.

The computation (time) complexity of the value determination step of policy iteration is a function of the size, S, of the state space. Traditional Gauss elimination has complexity $O(S^3)$. 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.

4.2.2. Policy improvement. The policy improvement step for link s consists of finding the action that maximizes the relative value in each state $\mathbf{x}\in X$:

$$a=\underset{u\in A(\mathbf{x})}{\operatorname{argmax}}\left\{R^s(\mathbf{x},u)-\overline{R}^s(\pi)\tau(\mathbf{x},u)+\sum_{y\in X}p_{xy}(u)v^s(\mathbf{y},\pi)\right\} \tag{41}$$

where $A(\mathbf{x})$ 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 $O(2^G S)$, where G denotes the number of unique bandwidth categories.

5. NUMERICAL RESULTS

5.1. Considered routing algorithms

The routing algorithms that are considered in the numerical experiments can be classified into MDP based routing algorithms and conventional routing algorithms. Eight MDP based routing algorithms are compared:

- *MDP*: MDP routing by reward maximization based on exact link model [7],
- *MDP_P*: MDP method with priority for the shortest path,
- *MDP_K*: MDP routing by cost minimization using Krishnan's and Hübner's state aggregation method [11] with original link reward parameters defined by Equation (3),
- *MDP_K+*: MDP_K routing with modified link reward parameters defined by Equation (26),
- *MDP_N+*: The new MDP routing framework based on reward maximization and modified reward parameters, proposed in Section 3.3,
- *MDP_PN+*: MDP_N+ method with priority for the shortest path,
- *MDP_H*: MDP routing by cost minimization using Hwang's, Kurose's and Towsley's state aggregation scheme [9],
- *MDP_PH*: MDP_H method with priority for the shortest path.

One conventional routing method, the LLR method, is included in the evaluation. The reason for choosing LLR is that it is among the routing methods with best performance [1, 4]. 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.

The LLR algorithm works as follows. When a class j call request is received, the set of shortest paths W_j^n with n links is considered first. Among this set, a path with largest free capacity of the bottleneck link greater than or equal to the bandwidth requirement b_j and greater than the trunk reservation value θ_j^n is searched for. The bottleneck link is the link with least free capacity along the path. Note that a unique trunk reservation value is used for every call from class j that is offered to the set of shortest paths of length n. If all paths have insufficient free bandwidth, the call is offered the next set of (longer) shortest paths, and the routing procedure is repeated. The procedure stops when a feasible path among the set of shortest paths is found, or when no path between the OD pair offers sufficient free bandwidth.

The priority mechanism for MDP routing works as follows. The set of shortest paths with sufficient free capacity is considered first. The call is offered to the path with largest positive path-net gain among this set. If no path has positive path-net gain, the next set of (longer) shortest paths is considered. The procedure stops when a feasible path with maximum positive path net-gain is found among the set of shortest paths, or when no path between the OD pair offers sufficient free bandwidth.

Table 1. Description of network example.

	W6N	W13N
Symmetrical	No	No
Number of nodes	6	13
Number of bi-directional links	15	21
Number of OD pairs	30	156
Number of routes per OD pair	5	1–7
Link capacity [Mbps]	12–192	50–100
Network capacity [Mbps]	2484	2400
Maximum number of links in path	2	4
Number of traffic categories	2	2
Mean holding time [s]	1, 10	1, 10
Bandwidth b_i [Mbps]	1, 6	1, 6
Network traffic [Mbps*Erlang]	1816.8	624.0
$r_j' = r_j\mu_j/b_j$	1	1

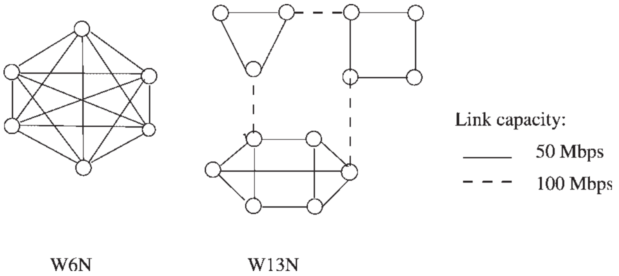

Figure 4. Network examples W6N and W13N.

5.2. Examples and results

The performance analysis is performed for the network examples W6N and W13N described in Table 1. The topologies of W6N and W13N is based on an example in References [2, 12] respectively. The topologies are shown in Figure 4. The link capacities and offered traffic volumes for network example W6N are based on the example in Reference [2] and is shown in Table 2. Network W6N corresponds to an STM type network, while network W13N corresponds to an ATM type network. The OD pairs in W6N are offered different traffic volumes (asymmetric case), while the OD pairs in W13N are offered the same traffic volumes (symmetric case). The algorithm specific parameter settings, presented in Table 3, were determined heuristically based on simulation experience.

Each curve in the diagrams contains N simulation points, $\bar{x}_k, k = 1, \ldots, N$, which are obtained as averages over M simulation runs per point: $\bar{x}_k = \frac{1}{M}\sum_{i=1}^{M} x_{ik}$. For assessment of the accuracy of the simulation results, we present values of the pooled standard deviation of simulation results in Tables 4 and 5 respectively. We compute

Table 2. Link capacity and offered traffic for 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 3. Algorithm specific parameters.

MDP adaptation epochs	6
MDP_K adaptation epochs	4
MDP_K + adaptation epochs	4
MDP_N + adaptation epochs	4
MDP_H 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
Number of simulation points per curve N	19
Number of simulation runs per point M	20
Hwang's heuristic parameter α_1	0
Hwang's heuristic parameter α_2	0.1
Trunk reservation parameter θ_j^n	0
Filtering probability $\phi_j^s(\mathbf{x}, \pi)$	1.0

Table 4. Pooled standard deviation in simulations with variable traffic ratio.

	W6N pooled reward loss SD (%)	W13N pooled reward loss SD (%)
MDP	0.65	0.03
MDP_P	0.32	0.03
MDP_K	0.03	0.02
MDP_K+	0.03	0.03
MDP_N+	0.03	0.03
MDP_PN+	0.03	0.03
MDP_H	0.06	0.03
MDP_PH	0.05	0.04
LLR	0.04	0.04

Table 6. CPU time for one simulation run in one point in simulations with variable traffic ratio.

	W6N simulation CPU time average (SD) (s)	W13N simulation CPU time average (SD) (s)
MDP	161.0 (6.4)	96.5 (2.0)
MDP_P	151.8 (2.4)	71.9 (0.9)
MDP_K	45.2 (0.2)	49.6 (0.1)
MDP_K+	45.7 (0.4)	49.3 (0.1)
MDP_N+	45.1 (0.1)	50.0 (0.1)
MDP_PN+	37.5 (0.1)	39.1 (0.1)
MDP_H	45.6 (0.1)	49.6 (0.1)
MDP_PH	37.6 (0.1)	40.9 (0.1)
LLR	8.9 (0.1)	9.2 (0.1)

the pooled standard deviation over the N simulation points as:

$$s = \sqrt{\frac{1}{N}\sum_{k=1}^{N} s_k^2} \tag{42}$$

where s_k^2 denotes the sample variance of the value of point k:

$$s_k^2 = \frac{1}{M-1}\sum_{i=1}^{M}(x_{ik} - \bar{x}_k)^2 \tag{43}$$

Table 6 shows the average simulation time and pooled standard deviation for one of M simulation runs carried out for each of the N simulation points per curve. The table is based on CPU time measurements for the first set of figures. One simulation run consists of an initial 'warm up' period, followed by a number of adaptation periods in case MDP routing is used, and finally a measurement period. Each adaptation period consists of a measurement period followed by a policy iteration step.

Figures 5–12 show the reward loss, $L = 1 - \overline{R}/R$, for the complete set of routing algorithms as a function of the traffic ratio. The ratio of OD-pair traffic is measured by $b_n\lambda_n\mu_n^{-1}/b_w\lambda_w\mu_w^{-1}$. Different mixes are obtained by varying the per-category call arrival rate to the OD pairs between the simulations, while keeping the amount of

Table 5. Pooled standard deviation in simulations with variable WB normalized reward parameter.

	W6N pooled NB blocking probability SD (%)	W6N pooled WB blocking probability SD (%)	W13N pooled NB blocking probability SD (%)	W13N pooled WB blocking probability SD (%)
MDP	2.1	1.2	0.2	0.1
MDP_N+	0.0	0.0	0.0	0.0
MDP_H	0.1	0.2	0.0	0.1

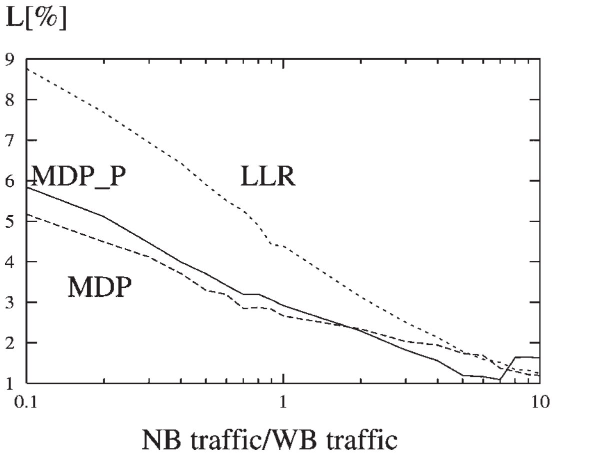

Figure 5. Reward loss versus traffic ratio for network W6N (reference methods).

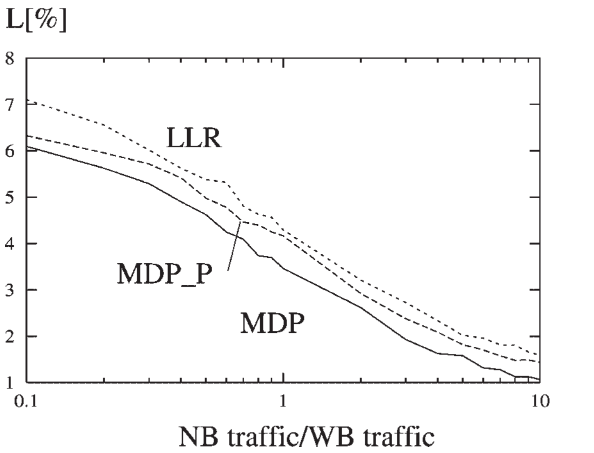

Figure 6. Reward loss versus traffic ratio for network W13N (reference methods).

traffic per OD pair constant. All OD pairs were offered the same per-category call arrival rates within a simulation.

Figures 13–16 show the per-category call blocking probability as a function of the normalized WB reward parameter for a restricted set of routing algorithms. The normalized reward parameter, r'_j, for class j fulfills $r'_j = r_j\mu_j/b_j$. We present the best results, which are obtained by varying the r'_{WB} parameter while keeping $r'_{NB} = 1$ for networks W6N and W13N. In the simulations, we have assumed a traffic ratio of 1.0, measured by $b_n\lambda_n\mu_n^{-1}/b_w\lambda_w\mu_w^{-1}$, for each OD pair.

5.3. Results analysis

From the graphs in Figures 5–12, the following conclusions are drawn:

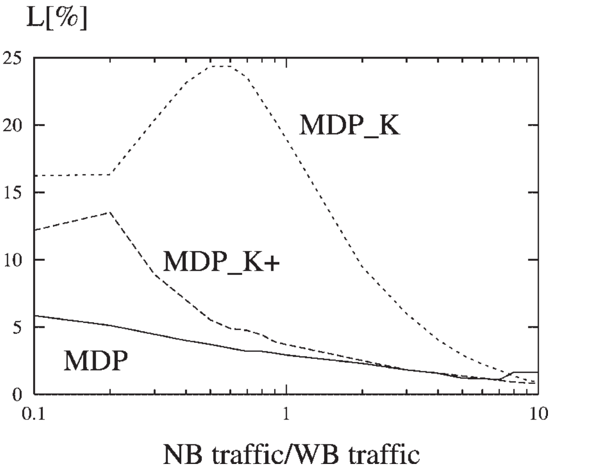

Figure 7. Reward loss versus traffic ratio for network W6N (Krishnan's method).

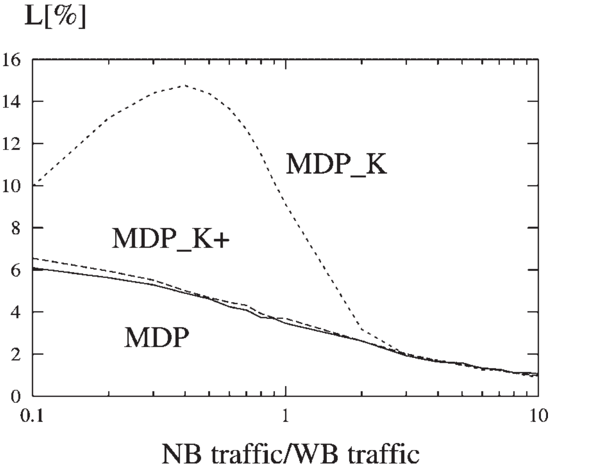

Figure 8. Reward loss versus traffic ratio for network W13N (Krishnan's method).

- The exact MDP method has the lowest reward loss.
- The LLR method has relatively high reward loss.
- The MDP_K method has relatively high reward loss.
- The MDP_K+ method has reward loss close to the exact MDP method, except for network W6N when WB traffic dominates.
- The MDP_N+ method has identical reward loss as the MDP_K+ method.
- The MDP_H method has higher reward loss than the MDP_N+ and MDP_K+ methods, except for network W6N when WB traffic dominates.
- The MDP priority mechanism is beneficial for W6N when WB traffic dominates but not for W13N in case the MDP_N+, MDP_K+ or MDP_H method is used.

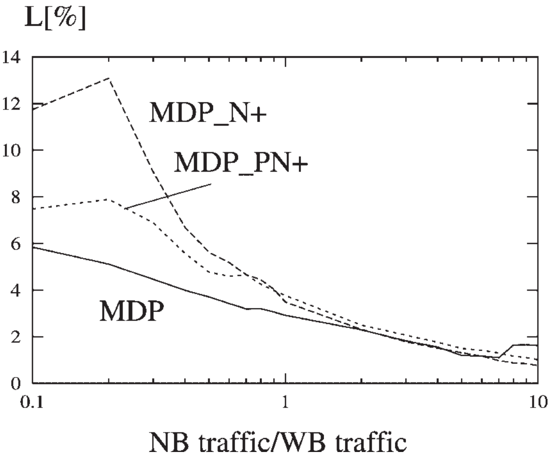

Figure 9. Reward loss versus traffic ratio for network W6N (new method).

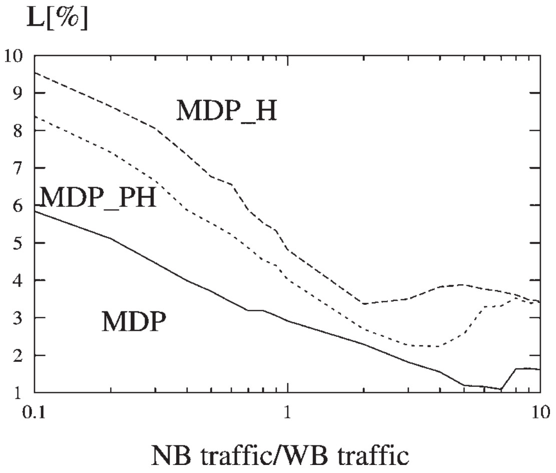

Figure 11. Reward loss versus traffic ratio for network W6N (Hwang's method).

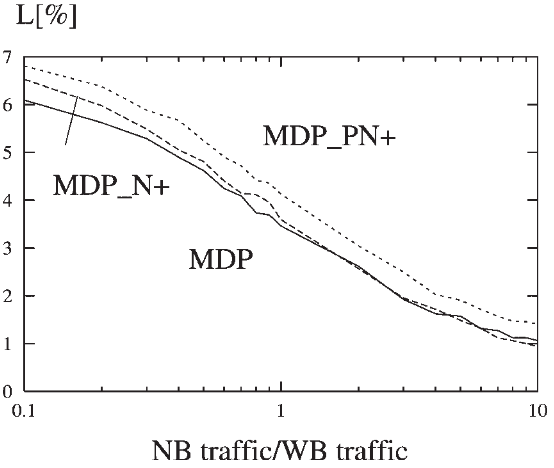

Figure 10. Reward loss versus traffic ratio for network W13N (new method).

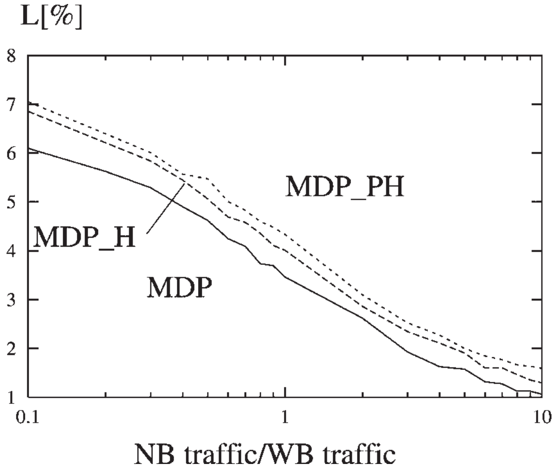

Figure 12. Reward loss versus traffic ratio for network W13N (Hwang's method).

- The MDP priority mechanism is not beneficial in case the exact MDP method is used.

From the graphs in Figures 13–16, the following conclusions are drawn:

- The ratio between the NB and WB call blocking probability can be controlled by varying the normalized WB reward parameter provided we use the MDP or MDP_N+ methods.
- The ratio between the NB and WB call blocking probability is not sensitive to changes in the normalized WB reward parameter provided we use the MDP_H method.

6. CONCLUSION

An optimal solution of the CAC and routing problem in multi-service loss networks, in terms of average reward per time unit, is possible by modeling the behavior of the network as a MDP. However, even after applying the standard link independence assumption, the solution of the corresponding set of link problems may involve considerable numerical computation.

In this paper, we studied an approximate MDP framework on the link level, where the vector-valued MDP states are mapped into a set of aggregate scalar MDP states

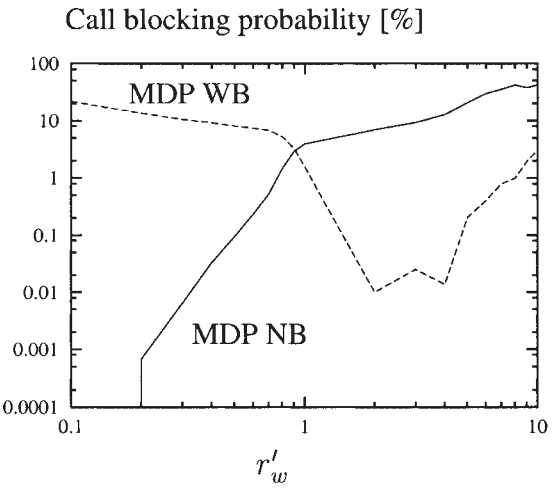

Figure 13. NB and WB call blocking probabilities versus normalized WB reward parameter for W6N (reference MDP method).

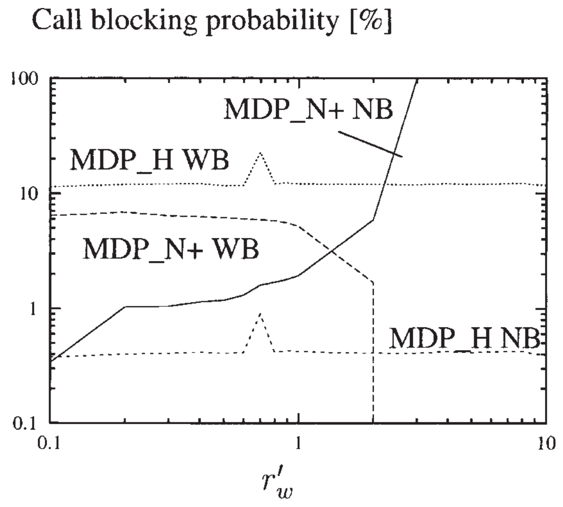

Figure 15. NB and WB call blocking probabilities versus normalized WB reward parameter for W6N (state aggregation MDP methods).

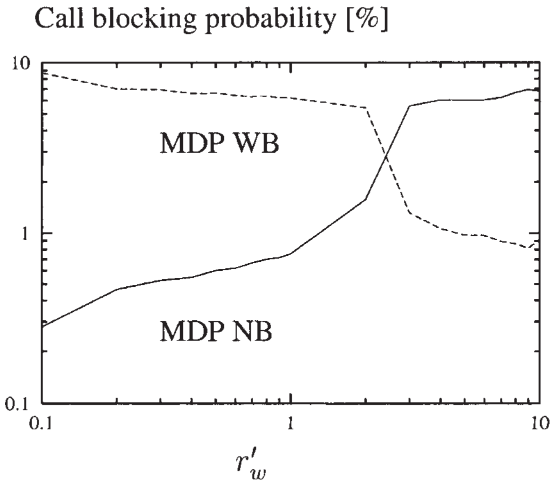

Figure 14. NB and WB call blocking probabilities versus normalized WB reward parameter for W13N (reference MDP method).

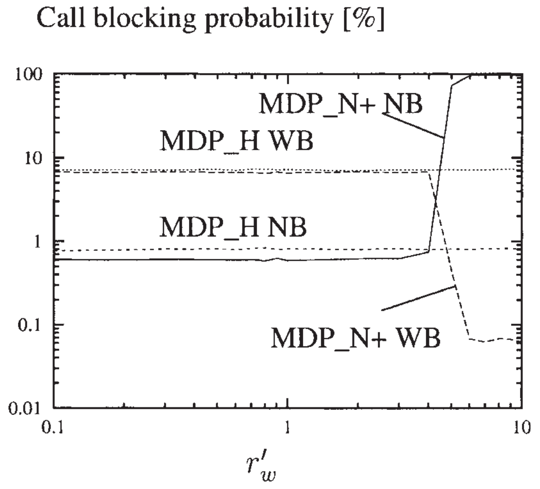

Figure 16. NB and WB call blocking probabilities versus normalized WB reward parameter for W13N (state aggregation MDP methods).

corresponding to link occupancies. In particular, we proposed a new model of the expected reward for admitting a call on the network. Compared to Krishnan's and Hübner's method [11], our reward model more accurately reflects the bandwidth occupancy by different call categories.

The new reward model is matched to the bandwidth sharing model in the state aggregation framework. In the new reward model, each reward-generating bandwidth unit is shared, from a conceptual viewpoint, between all call categories present on the link. In the old reward model, each bandwidth unit is only occupied by one call category. As a result, the new reward model enables a more accurate modeling of the path net-gain values, and forms the basis for more efficient CAC and routing decisions.

The exact and approximate link MDP frameworks were compared by simulations. The results showed that the proposed link reward model significantly improves the performance of MDP-based CAC and routing with state aggregation.

REFERENCES

1. Ash G. *Dynamic Routing in Telecommunication 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. Chung S, Ross K. Reduced load approximations for multirate loss networks. *IEEE Transactions on Communications* 1993; **41**(8): 1222–1231.
4. Dziong Z. *ATM Network Resource Management*. McGraw-Hill (ISBN 0-07-018546-8), New York, 1997.
5. Dziong Z, Liao K-Q, Mason LG. Flow control models for multi-service networks with delayed call set up. In *Proceedings of IEEE INFOCOM'90*, IEEE Computer Society Press, 1990; pp. 39–46.
6. Dziong Z, Liao K-Q, Mason LG, Tetreault N. Bandwidth management in ATM networks. In *Proceedings of ITC'13*, Jensen A, Iversen VB (eds). 1991; pp. 821–827.
7. Dziong Z, Mason L. Call admission and routing in multi-service loss networks. *IEEE Transactions on Communications* 1994; **42**(2): 2011–2022.
8. Dziong Z, Mignault J, Rosenberg C. Blocking evaluation for networks with reward maximization routing. In *Proceedings of INFOCOM'93*, San Francisco, USA, 1993.
9. Hwang R, Kurose J, Towsley D. State-dependent routing for multirate loss networks. In *Proceedings of Globecom'92*, Orlando, 1992; pp. 740–747.
10. Kaufman J. Blocking in a shared resource environment. *IEEE Transactions on Communications*, 1981; **COM-29**(10):1474–1481.
11. Krishnan K, Hübner F. Admission control and routing for multirate circuit-switched traffic. In *Proceedings of ITC'15*, Washington, USA, 1997.
12. Ma Q, Steenkiste P, Zhang H. Routing high-bandwidth traffic in max–min fair share networks. In *Proceedings of ACM SIGCOMM'96*, Stanford, CA, USA, 1996; pp. 206–217.
13. Marbach P, Mihatsch O, Tsitsiklis J. Call admission control and routing in integrated services networks using neuro-dynamic programming. *IEEE Transactions on Communications* 2000; **18**(2):197–208.
14. Nordström E. Near optimal link allocation of blockable narrow-band and queueable wide-band call traffic in ATM networks. In *Proceedings of ITC'15*, Washington, 1997.
15. Nordström E, Dziong Z. CAC and routing in multi-service networks with blocked wide-band calls delayed—part I, exact link MDP framework, submitted, 2004.
16. Nordström E, Dziong Z. CAC and routing in multi-service networks with blocked wide-band calls delayed—part II, approximate link MDP framework, submitted, 2004.
17. Roberts J. A service system with heterogeneous user requirements. In *Performance of Data Communications Systems and Their Applications*, Pujolle G (ed.). North-Holland, Amsterdam, 1981.
18. 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.
19. Tijms H. *Stochastic Modeling and Analysis—a Computational Approach*. Wiley: New York, 1986.

AUTHORS' BIOGRAPHIES

Ernst Nordström is a senior lecturer at the Culture, Media and Computer Science Department at Dalarna University, Sweden. His research interests include Markov decision process theory and optimization theory with application to CAC, routing and dimensioning in multi-service networks. He did his Ph.D. in Computer Systems in 1998 and his M.Sc. in Engineering Physics in 1992, both from Uppsala University, Sweden.

Jakob Carlström is a senior systems architect at Xelerated, where his research and development work focuses on circuits and systems for network processing. Carlström's interests include algorithms and architectures for network control, packet forwarding and resource allocation. He did his Ph.D. in Computer Systems in 2000 and his M.Sc. in Engineering Physics in 1994, both from Uppsala University, Sweden.