Near-Optimal CAC and Routing for Multi-Service Networks by Markov Decision Theory

Ernst Nordström

Department of Computer Science, Dalarna University

SE-781 88 Borlänge, Sweden,

Email: eno@du.se

Abstract

In this paper we study the call admission control (CAC) and routing issue in multi-service networks. Two categories of calls are considered: a wide-band with delayed call set up and a narrow-band operating in loss mode. The control objective is formulated as reward maximization with penalty for delay. A suboptimal solution is acheived by applying Markov decision process (MDP) theory together with a two-level approximation. First, the network is decomposed into a set of links assumed to have independent traffic and reward processes, respectively. Second, by applying state aggregation or decomposition of the link Markov process, the link MDP tasks are simpified considerably. The numerical results show that the MDP-based methods are able to find an efficient trade-off between reward loss and average call set up delay, outperforming the conventional Least Loaded Path (LLP) method.

1 INTRODUCTION

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. First, the network should determine the set of feasible paths between the source and destination which offers sufficient QOS to the new and existing calls in terms of delay, jitter and data loss. Second, the network should chose to reject the call or to accept it on some path among the set of feasible paths. This choice should be consistent with GOS contraints, in terms of call blocking probabilities and call set up delays, as well as maximizing the average revenue rate for the operator.

Two categories of calls are considered: a narrow-band (NB) requesting a bandwidth of b_n Mbps, and wide-band (WB) requesting b_w Mbps $(b_n < b_w)$. The required bandwidth is represented by the call's peak bandwidth in case of deterministic multiplexeing, and by the call's effective bandwidth in case of statistical multiplexing. 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. There exists two main approaches to cope with this fairness problem: access control of NB calls or queueing 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 acheived at the expense of bandwidth utilization.

Queueing of a WB call is done when the network is found busy by the WB call arrival. 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 uilization at the same time.

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. 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. State-dependent mechanims offer advantages both in terms of acheivable revenue and ability to control the QOS and GOS.

This paper deals with a particular form of state-dependent CAC and routing, where the behaviour of the network is formulated as Markov Decision Process (MDP) [1, 11] . A MDP is a controlled Markov process, where the set of state transistions from the current Markov

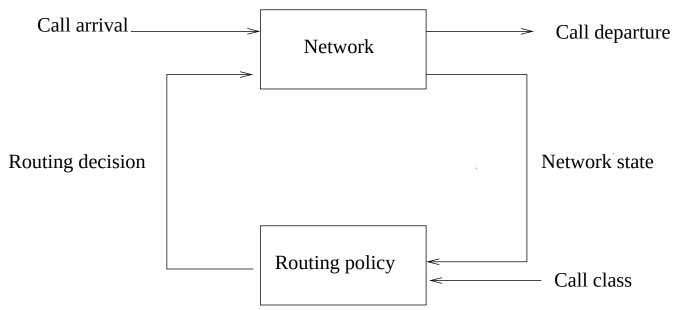


Figure 1: State-dependent routing

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 reward parameter and the objective is to maximize the reward from carried calls. The delay of WB calls is taken into account by adding a penalty term to the objective function. The penalty term is a function of the average delay.

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. Therefore, a necessary modelling 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, or when the maximal queue size increases. In the first case, the increase is exponential, in the second and third case it is polynomial. 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 and decomposition of the link Markov process. State aggregation can not be used between the NB loss category and the WB delay category, since the state space is not coordinate convex. However, we can use state aggregation to form a NB loss super-category from loss sub-categories with different NB bandwidth requirements. Link Markov process decomposition can then be applied between the NB super-category and the WB category.

The state aggregation methods include the work by Krishnan and Hübner [7]. They proposed a simplified framework based on a scalar link state representing the link occupancy. Transition probabilities between link states were derived from link occupancy probabilities obtained by a recursive procedure due to Kaufman [6] and Roberts [10]. The MDP task was solved by one-step policy iteration. A modification of the link reward model used in Krishnan's and Hübner's method was proposed by Nordström and Carlström [9]. The modification can improve the network revenue significantly according to simulation results in [9].

The link Markov process decomposition method is due to Liao and Mason [5] and Dziong, Liao and Mason [2]. 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 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 acount. The decomposition method for networks operating under delayed call set up is further developped by Dziong and Nordström in [4].

This paper provides an numerical/simulation-based evaluation of the exact and approximate MDP frameworks in terms of computational time complexity, average reward loss and average call set up delay. For comparison, the performance of a conventional routing method, the Least Loaded Path (LLP) method, is also evaluated.

The paper is organized as follows. Section 2 formulates the CAC and routing problem in terms of offered traffic, network and queueing model, and optimization objective. Section 3 describes the network and exact link MDP models. Section 4 outlines the MDP computation procedure. Section 5 presents the size of the state space for the exact and approximate link MDP models. Section 6 give a summary of the numerical/simulation-based evaluation of the performance of the exact and approximate link MDP models as well as the LLP routing method. Finally, Section 7 concludes the paper.

2 PROBLEM FORMULATION

2.1 Traffic assumptions

The network is offered traffic from K classes. The j-th class is characterized by the following:

- origin-destination (OD) node pair,
- bandwidth requirement $b_j \in \{b_n, b_w\}$ [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 $G = 2$ bandwidth categories. The i-th category is characterized by:

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

2.2 Network and queueing model

The network is assumed to consist of a set of switching nodes, interconnected by links according to some network topology. Each interconnection consists of two links, carrying traffic in opposite directions. Each link has one finite FIFO queue for WB call requests. The queues operates as follows. 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 some link along the path is not able to directly accept the new WB call. At those links, join the new WB call request at the tail of the queue, provided the queue is not held at its maximum. 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, reserve bandwidth 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.

The advantage of this scheme is its simplicity but its performance may have some drawbacks. One is that the ”reservation” traffic caused by multilink calls due to bandwidth reservation on some links while the call request is in the queue of other links. Although this ”reservation” traffic is likely to be negligible under nominal conditions, it can be significant in case of overloads. Another, less critical, drawback of this solution is the lack of access fairness since the average waiting time of calls offered to multi-link paths is a sum of average waithing times of all queues.

2.3 Objective function

In our loss-delay type of systems we have to deal with bi-objective type function. Let us first define each objective separately. To take into account traffic losses due to rejection of NB and WB calls we apply reward formulation. In this case the reward from a carried call is defined by the reward rate $q_j = r_j \mu_j$ where r_j, μ_j denotes the reward parameter and the departure rate of the j-th type call, respectively. Now we can define the objective function, $\overline{R}$, as average reward from the network given by

$$\overline{R} = \sum_j r_j \overline{\lambda}_j \tag{1}$$

where $\overline{\lambda}_j$ denotes the j-th class call acceptance rate (the process is assumed to be stationary). The obvious goal of the CAC and routing algorithm is to maximize the objective function. This approach was already applied in state-dependent routing schemes for loss systems presented in [3, 1]. The reward maximization has several advantages from the management point of view since by controlling the reward parameters one can almost independently control the GOS of individual streams (cf. [3, 1]). Another advantage is that by using the control model presented in [3, 1], the objective function can be decomposed as:

$$\overline{R} = \sum_s \overline{R}^s \tag{2}$$

where $\overline{R}^s$ denotes the average reward from the s-th link. Since in our system the NB and WB calls are treated differently it may be convenient to separate the corresponding rewards:

$$\overline{R} = \overline{R}_n + \overline{R}_w = \sum_s [\overline{R}_n^s + \overline{R}_w^s] \tag{3}$$

If we would be concerned only with the delay of WB calls, the natural objective of the CAC and routing algorithm would be to minimize the average delay of calls, $\overline{D}$. For the proposed management of queueing system such an objective function is given by

$$\overline{D} = \sum_s \overline{D}^s \frac{\lambda_w^s}{\lambda_w} \tag{4}$$

where $\overline{D}^s$ denotes the average delay of calls in the s-th link queue andλ_w^s, and λ_w denotes the arrival rate of WB calls offered to the s-th link and the network, respectively.

We select for the objective function a linear combination of the two objective functions, which can also be interpreted as reward maximization with penalty for delay of WB calls:

$$\overline{R}_D = \overline{R} - \alpha \overline{D} \tag{5}$$

where α is the delay penalty weight which determines the trade-off value between the reward and average delay. By using (2),(4) in (5) we arrive at

$$\overline{R}_D = \sum_s [\overline{R}^s - \alpha \overline{D}^s \frac{\lambda_w^s}{\lambda_w}] \tag{6}$$

This form of the objective function illustrates the desired separability of the objective function.

Note that the form of (6) suggests that the delay penalty weight can be also link dependent:

$$\overline{R}_D = \sum_s [\overline{R}^s - \alpha^s \overline{D}^s \frac{\lambda_w^s}{\lambda_w}] \tag{7}$$

This feature gives additional freedom of distributing the delay among the links which may be of importance in practical problems.

3 MDP MODELLING

3.1 Network decomposition

In the exact network MDP model, the network state is given by the matrix $\mathbf{y} = \{x^s\}, s \in S$, where x^s denotes the state of link s and S denotes the set of all link indices in the network. The action space is given by

$$A = \{a = \{a_j\} : a_j \in \{0\} \cup W_j, j \in J\} \tag{8}$$

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.

In the exact MDP framework, the network state and the 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 [3].

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 π denotes the CAC and routing policy. In particular, a call connected on a path constisting 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 of separable link reward processes. The link call reward parameters $r_j^s(\pi)$ fulfill the obvious condition that

$$r_j = \sum_{s \in S_k} r_j^s(\pi) \tag{9}$$

where S_k denotes the set of links consistuting path k, specified by the routing policy π. Different models for computing link reward parameters are possible [3]. In this paper we use a simple rule: the link reward 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 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 $i \in I = \{1, ..., G\}$ with average reward parameter defined as [3]:

$$\overline{r}_i^s(\pi) = \frac{\sum_{j \in J_i} r_j^s(\pi) \overline{\lambda}_j^s(\pi)}{\sum_{j \in J_i} \overline{\lambda}_j^s(\pi)} \tag{10}$$

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.

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 $\mathbf{x} = (x_n, x'_w)$, where x_n denotes the number of NB calls on the link, and x'_w denotes the number of WB calls in the system (on the link and in the queue). The state space X for the exact link model is given by:

$$X = \{\{\mathbf{x} = (x_n, x'_w) : 0 \leq x_n \leq N^s_n, 0 \leq x'_w \leq N^s_w + L^s, \\ x_l = f(\mathbf{x}), x_n b_n + (x'_w - x_l) b_w \leq C^s\} \tag{11}$$

where $N^s_n = \lfloor C^s / b_n \rfloor$, $N^s_w = \lfloor C^s / b_w \rfloor$, and C^s, L^s, denotes the capacity and maximal size of link and queue s, respectively. The number of WB calls in the queue, x_l, is obtained from state $\mathbf{x}$ as follows:

$$x_l = f(\mathbf{x}) := \inf\{x_l : C^s - x_n b_n \geq (x'_w - x_l) b_w\} \tag{12}$$

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:

$$A = \{a = \{a_i\} : a_i \in \{0, 1\}, i \in I\} \tag{13}$$

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:

$$A(x) = \{a \in A : a_i = 0 \text{ if } \mathbf{x} + \delta_i \notin X, i \in I\} \tag{14}$$

where δ_i denotes a vector of zeros exept 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:

$$p_{xy}(a) = \begin{cases} \lambda_i^s(\mathbf{x},\pi) a_i \tau(\mathbf{x},a), & \mathbf{y} = \mathbf{x} + \delta_i \in X, i \in I \\ x_n \overline{\mu}_n \tau(\mathbf{x},a), & \mathbf{y} = \mathbf{x} - \delta_n \in X, \\ (x'_w - x_l)\overline{\mu}_w \tau(\mathbf{x},a), & \mathbf{y} = \mathbf{x} - \delta_w \in X, \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

where $\lambda_i^s(\mathbf{x},\pi)$ denotes the i-th category arrival rate to the link in state $\mathbf{x}$ under routing policy π. The parameters $\overline{\mu}_n$, $\overline{\mu}_w$ denotes the departure rate of NB and WB calls, respectively, 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:

$$\lambda_i^s(\mathbf{x},\pi) = \sum_{j\in J_i} \lambda_j^k(\pi)\phi_j^s(\mathbf{x},\pi) \prod_{c\in k\setminus\{s\}} (1 - B_j^c(\pi)), \tag{16}$$

where $B_j^c(\pi)$ denotes the probability that link c has not enough capacity to accept a class j call, $\phi_j^s(\mathbf{x},\pi)$ denotes a filtering probability that the path net-gain is positive. The filtering probability can be computed using link state distributions, or approximated with one according to experiments in [3]. The $\lambda_j^k(\pi)$ denote the arrival rate of class j to path $k \in W_j$, and is given the following load sharing model [3]:

$$\lambda_j^k(\pi) = \lambda_j \frac{\overline{\lambda}_j^k(\pi)}{\sum_{h\in W_j} \overline{\lambda}_j^h(\pi)} \tag{17}$$

where the $\overline{\lambda}_j^k(\pi)$ 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 NB category is computed as:

$$\overline{\mu}_n = \left[\sum_{j\in J_n} p_j^n \mu_j^{-1}\right]^{-1} \tag{18}$$

where p_j^n denotes the probability that an arbitrary NB call found on the link is from class $j \in J_n$:

$$p_j^n = \frac{\overline{\lambda}_j^s(\pi)}{\sum_{c\in J_n} \overline{\lambda}_c^s(\pi)} \tag{19}$$

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

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

$$\overline{\mu}_w = \left[\sum_{j \in J_w} p_j^w \mu_j^{-1} + T^s \right]^{-1} \tag{20}$$

where T^s denotes the average reservation time for WB calls on link s (T^s can be obtained from measurements), and p_j^w denotes the probability that an arbitrary WB call found on the link is from class $j \in J_w$.

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

$$\tau(\mathbf{x}, a) = \left\{ \sum_{i \in I} x_i \overline{\mu}_i + a_i \lambda_i^s(\mathbf{x}, \pi) \right\}^{-1} \tag{21}$$

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

$$q^s(\mathbf{x}) = \overline{r}_n^s x_n \overline{\mu}_n + \overline{r}_w^s (x'_w - x_l)\overline{\mu}_w - \alpha^s \frac{x_l}{\lambda_w} \tag{22}$$

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 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 chose, given the state of the network and the class of the call request, a path which offer maximal positive path net-gain among the paths with sufficient QOS (See Figure 2). The call is rejected if the path net-gain is negative, or if no path would offer sufficient QOS.

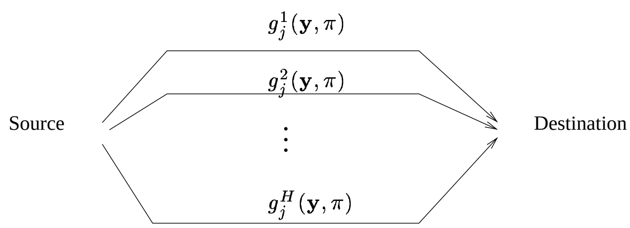


Figure 2: The call is offered to a path which has sufficient QOS and maximal positive path net-gain among the $H = |W_j|$ alternative paths.

The path net-gain is defined as:

$$g_j^k(\mathbf{y}, \pi) = r_j - \sum_{s \in S_k} p_i^s(\mathbf{x}, \pi) \tag{23}$$

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

$$p_i^s(\mathbf{x}, \pi) = \overline{r}_i^s(\pi) - g_i^s(\mathbf{x}, \pi) \tag{24}$$

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 admisson of a category-i call in link state $\mathbf{x}$ and is defined as follows:

$$g_i^s(\mathbf{x}, \pi) = v^s(\mathbf{x} + \delta_i, \pi) - v^s(\mathbf{x}, \pi) \tag{25}$$

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.

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

$$R_D^s(x_0, \pi, T) = E\left[\int_{t_0}^{t_0+T} q^s(\mathbf{x}(t))dt\right] \tag{26}$$

where $q^s(\mathbf{x}(t))$ denotes the exected 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:

$$v^s(\mathbf{x}_0, \pi) = \lim_{T \to \infty} [R_D^s(\mathbf{x}_0, \pi, T) - R_D^s(\mathbf{x}_r, \pi, T)] \tag{27}$$

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).

In the context of the exact link MDP model outlined in the previous section, 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 aceptance 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 $v^s(\mathbf{x}, \pi)$ for the current routing policy π

   (c) **Policy improvement**: Improve the link CAC policies π_s based on the new relative values

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 [11].

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:

$$\begin{cases} v^s(\mathbf{x}, \pi) = R_D^s(\mathbf{x}, a) - \overline{R}_D^s(\pi)\tau(\mathbf{x}, a) + \sum_{\mathbf{y} \in X} p_{xy}(a) v^s(\mathbf{y}, \pi) \\ v^s(\mathbf{x}_r, \pi) = 0; \qquad \mathbf{x} \in X, \end{cases} \tag{28}$$

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 $\mathbf{x}$
- $R_D^s(\mathbf{x}, a)$: the expected link reward when leaving state $\mathbf{x}$
- $p_{xy}(a)$: the transition probability from state $\mathbf{x}$ to state state $\mathbf{y}$, given that action a is taken in state 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
- $\overline{R}_D^s(\pi)$: 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.

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

$$a = \underset{a \in A(\mathbf{x})}{\mathsf{argmax}} \left\{ R_D^s(\mathbf{x}, a) - \overline{R}_D{}^s(\pi)\tau(\mathbf{x}, a) + \sum_{y \in X} p_{xy}(a) v^s(\mathbf{y}, \pi) \right\} \tag{29}$$

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 consitute an improved policy π'_s to be used again in the first step. After some simplification, the policy improvement step can be transformed into:

$$a = \underset{u \in A(\mathbf{x})}{\mathsf{argmax}} \left\{ \sum_{i \in I} u_i g_i^s(\mathbf{x}, \pi) \right\}, \mathbf{x} \in X \tag{30}$$

where $g_i^s(\mathbf{x}, \pi)$ is the link net-gain for category i in state $\mathbf{x}$. The policy improvement step has complexity O$(4GS)$, where G denotes the number of unique bandwidth categories.

5 STATE SPACE CARDINALITY

5.1 Exact link model

The size of the state space for the exact link model is given by:

$$S = \sum_{x'_w=0}^{N^s_w+L^s} N_n(x'_w) \tag{31}$$

where

$$N_n(x'_w) = C^s - (x'_w - x_l)b_w + 1, \tag{32}$$

where $x_l = f(\mathbf{x})$ denotes the number of WB calls in the queue in state $\mathbf{x}$. It can be shown that the size of the state space grows like:

$$S \sim \left\{ \frac{1}{G!} \prod_{i \in I} S_i \right\} + L^s \left\{ \lfloor C^s / b_n \rfloor + 1 \right\}, \tag{33}$$

where $S_n = N^s_n + 1$ and $S_w = N^s_w + 1$ denotes the maximal number of NB and WB states, respectively, on the the link when the other category is not present.

5.2 State aggregation link model

In the state aggregation method [7], the G-dimensional micro state $(x_1, ..., x_G)$ is aggregated into a one-dimensional macro state $m = \sum_{i \in I} b_i x_i$. The state space will contain $S = C^s + 1$ states, independent of the number of categories G [9]. This is under the assumption that the bandwidth requirements are integer valued, and at least one of the categories requires one bandwidth unit.

5.3 Decomposed link model

The NB link process is assumed to reach steady state distribution for each number of WB calls, x'_w, in the system (on the link and in the queue). The state of the NB link process can be described by a vector $\mathbf{x} = (x_n, x_w)$, where x_n and x_w denotes the number of NB and WB calls, respectively, on the link. A Markov NB decision problem is associated with each value x'_w of the number of WB calls in the system.

The size of the NB state space in WB state x'_w is given by [4]:

$$S_n(x'_w) = \sum_{\max(0, x'_w - L^s)}^{\min(n_w, x'_w)} N_n(x'_w, x_w) \tag{34}$$

where

$$N_n(x'_w, x_w) = \begin{cases} C^s - x_w b_w + 1, & x'_w = x_w \\ b_w, & x'_w > x_w, x_w < N^s_w \\ 1, & x'_w > x_w, x_w = N^s_w \end{cases} \tag{35}$$

The state of the WB link process can be described by a vector $\mathbf{x} = (x'_w, x_w)$, where x'_w and x_w are defined above. The size of the WB state space is given by [4]:

$$S_w = \sum_{x'_w = 0}^{N^s_w + L^s} N_w(x'_w) \tag{36}$$

where

$$N_w(x'_w) = \min(n_w, x'_w) - \max(0, x'_w - L^s) + 1 \tag{37}$$

6 NUMERICAL RESULTS

6.1 Considered routing algorithms

The routing algorithms that have been considered in the numerical experiments can be classified into MDP based routing algorithms and conventional routing algorithms. Three MDP based routing algorithms have been evaluated:

- MDP_E – MDP routing based on exact link model [4],
- MDP_A – MDP routing based on Krishnan's and Hübner's state aggregation link model [7] with modified link reward parameters [9].
- MDP_D – MDP routing based on decomposed link model [4],

For comparison, the performance of a conventional routing method, the Least Loaded Path (LLP) method, is also evaluated. The LLP routing method works as follows. For NB calls: the set of shortest paths, which has sufficient capacity to carry the new call, is considered first. Among these paths, the NB call is offered to the path with maximal available bandwidth. For WB calls: the set of shortest paths which requires no queueing is considered first. Among these paths, the WB call is offered to the path with largest available capacity. If queueing is necessary, the WB call is offered to the path which has the smallest maximal queue length along the path among the current set of shortest paths. For both NB and WB calls: if all shortest paths are busy, the set of the next shortest paths is investigated according to the same path selection principle.

6.2 Examples and results

The performance analysis is performed for the network example W16N described in Table 1. The algorithm specific parameter settings, presented in Table 6.2, were determined heuristically based on simulation experience. The topology of the network is shown in Figure 3.

The first set of figures shows the computational time complexity of one cycle of policy iteration for a single link in network example W16N. The value determination step is assumed to require O(S^3) operations, and the policy improvement step O($4GS$) operations, where S denotes the size of the state space. The cube root of the complexity as a function of the link capacity is shown in Figure 4 in the pure loss case $(L^s = 0)$. The cube root of the complexity as a function of the maximal queue size is shown in Figure 5. Note that approximate link model based on state aggregation (MDP_A) can only be used in the case with no queueing.

The second set of figures shows the routing performance (reward loss, average call set up delay, objective reward loss) in network example W16N as a function of the traffic mix. The mix of OD-pair traffic is measured by the ratio $b_n\lambda_n\mu_n^{-1}/b_w\lambda_w\mu_w^{-1}$. Different mixes are obtained by varying the per-category call arrival rate to the OD pairs between the simulations. All OD pairs were offered the same per-category call arrival rates within a simulation.

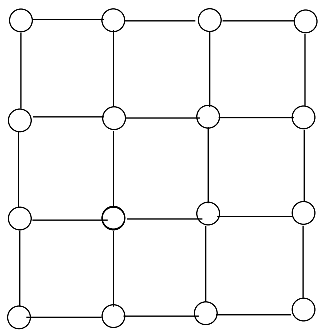
Figure 6 - 7, Figure 8, and Figure 9 show the reward loss, avarage call set up delay, objective reward loss, respectively, as a function of the traffic mix. The reward loss, average call set up delay and objective reward loss, respectively, are given by:

$$L = 1 - \overline{R}/R, \tag{38}$$

$$\overline{D} = \sum_s \overline{D}_s \frac{\lambda_w^s}{\lambda_w}, \tag{39}$$

$$L_D = 1 - \overline{R}_D / R. \tag{40}$$

The third set of figures shows the routing performance in network example W16N as function of the maximal queue size, assuming a traffic mix of 0.5 for each OD pair. Figure 10, 11, 12 and 13 show the reward loss, average call set up delay, objective reward loss, per-category call blocking probability, respectively, as a function of the maximal queue size.



W16N

Figure 3: Network example W16N

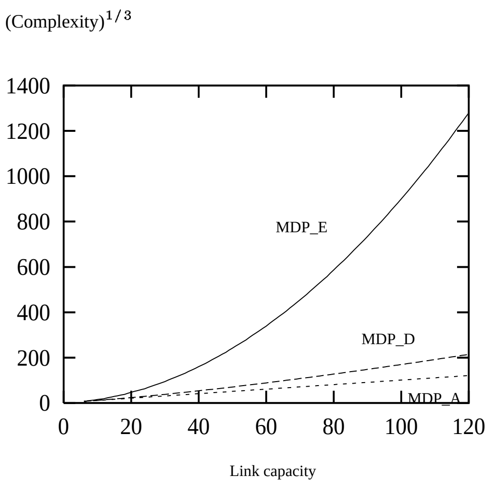


Figure 4: Cube root of the complexity of one policy iteration cycle as a function of the link capacity for the pure loss case

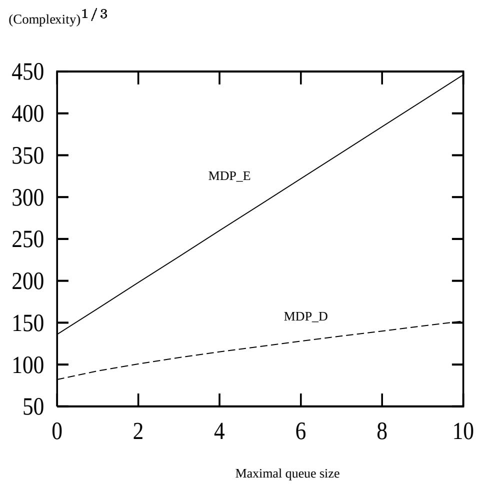


Figure 5: Cube root of the complexity of one policy iteration cycle as a function of the maximal queue size

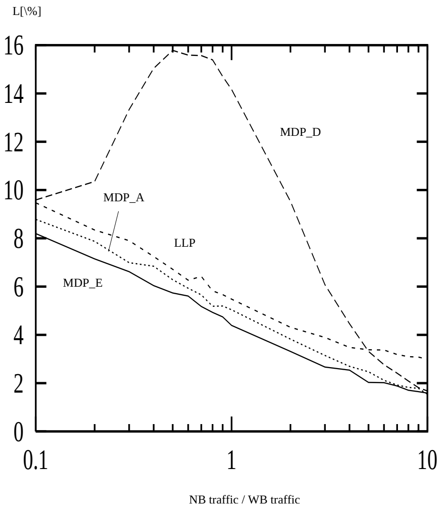


Figure 6: Reward loss versus traffic mix for network W16N with $L^s = 0$

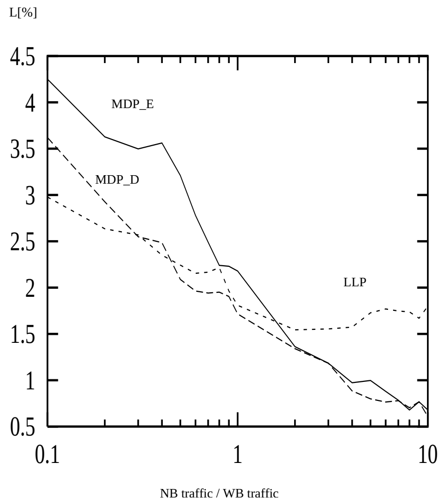


Figure 7: Reward loss as versus traffic mix for network W16N with $L^s = 3$

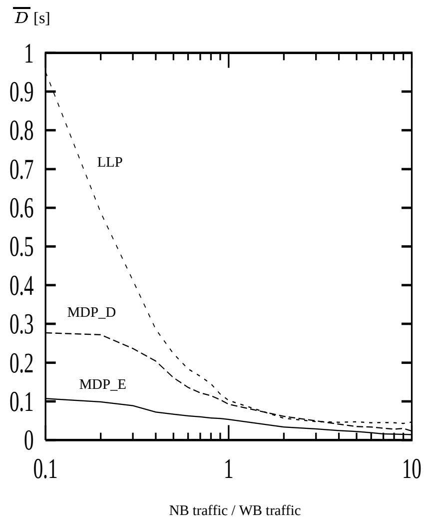


Figure 8: Average delay versus traffic mix for network W16N with $L^s = 3$

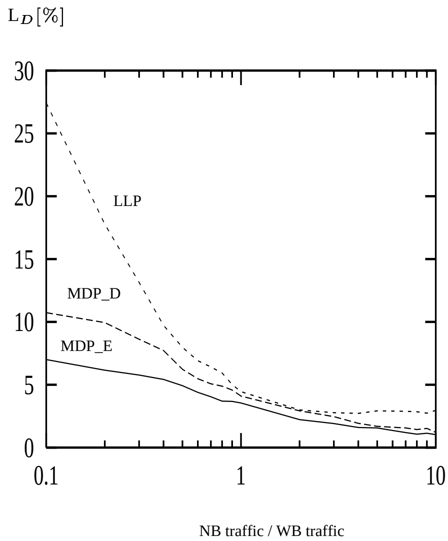


Figure 9: Objective reward loss versus traffic mix for network W16N with $L^s = 3$

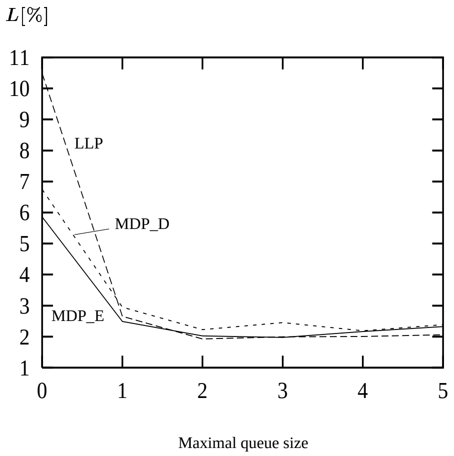


Figure 10: Reward loss versus maximal queue size for network W16N

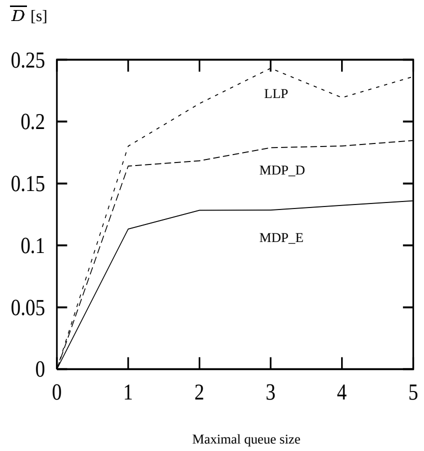


Figure 11: Average call set up delay versus maximal queue size for network W16N

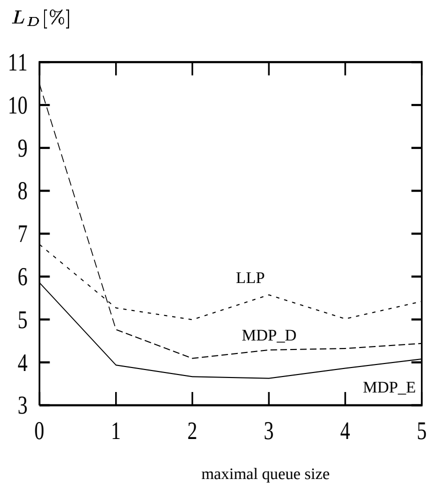


Figure 12: Objective reward loss versus maximal queue size for network W16N

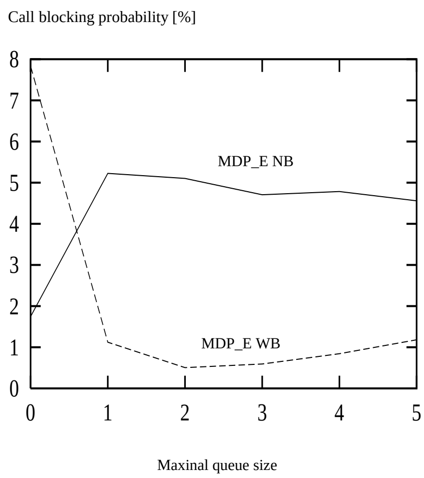


Figure 13: NB and WB call blocking probability with MDP_E for network W16N

	W16N
symmetrical	no
#nodes	16
#links	48
#OD pairs	240
#routes per OD pair	4-28
link capacity	60
network capacity	2880
maximal queue size	3
#max links in path	6
#traffic categories	2
mean holding time	1, 10
bandwidth b_i	1, 6
network traffic [Mbps*Erlang]	778
$r'_j = r_j \mu_j / b_j$	1

Table 1: Description of network example

	W16N
MDP_E adaptation epochs	6
MDP_D adaptation epochs	6
MDP_A adaptation epochs	3
call events in adaptation period	500 000
call events in measurement period	1000 000
#simulation runs per curve	20
delay penalty weight, Figure 6 - 9	200
delay penalty weight, Figure 10 - 12	100

Table 2: Algorithm specific parameters

6.3 Results Analysis

For the first set of figures which show the computational complexity the following conclusions are drawn:

- The difference in cube root of the computational complexity between the exact and approximate MDP models increases very fast when the link capacity increases (Figure 4).
- The cube root of the complexity grow linearly as a function of the maximal queue size, both for the exact (MDP_E) and approximate (MDP_D) link model (Figure 5).

Note that to obtain the true complexity the values in the curves should be raised by a power of 3.

For the second set of figures showing the routing performance for different traffic mixes, the following conclusions are drawn:

- The best reward loss in the pure loss network case is obtained for MDP_E followed by MDP_A, LLP and MDP_D, in that order (Figure 6).
- The MDP_D and especially MDP_E methods tend to reject many WB calls when WB traffic dominates in order to reduce the call set up delay (Figure 7).
- The average delay can be quite large for the LLP method when WB traffic dominates (Figure 8).
- The best overall behaviour for delayed call set up in terms of the objective function is obtained for MDP_E method, followed by the MDP_D and LLP method, respectively (Figure 9).

The MDP_D method has very bad performance in the pure loss case. The reason is that it fails to implement "intelligent blocking" of NB calls in favor for WB calls. Although the NB Markov state contain the number of WB calls, this variable will remain constant in the pure loss case, preventing intelligent blocking to be discovered. On other hand, in the delay call set up case, intelligent blocking of NB calls is possible, since the number of WB calls on the link will change when the queue is served.

For the third set of figures showing the routing performance for different maximal queue sizes, the following conclusions are drawn:

- Call queueing reduces the reward loss (Figure 10).
- The average call set up delay increases when the maximal queue size increases (Figure 11).
- The best overall behaviour in terms of the objective function is obtained for MDP_E method, followed by the MDP_D and LLP method, respectively (Figure 12).
- In a pure loss network the WB call blocking probabiliy is much higher than for the NB category. By using WB call queues with maximal size of one or more, the situation is reversed (Figure 13).

Note the trade-off between call set up delay and avarage reward loss, determined by the MDP-based methods, is done for a given set of the delay penalty weights α_s.

7 CONCLUSION

In this paper we formulated the CAC and routing problem as a reward maximization problem with penalty for WB call set up delay. In this formulation each call class is characterized by its reward parameter defining the expected reward for carrying a call from this class. Such a formulation allows to apply Markov Decision Process (MDP) theory to solve the problem. To make the solution feasible we decomposed the network into a set of links assumed to have independent traffic and reward processes, respectively.

The computational burden of each link MDP task increases exponentially when the number of categories increases, and polynomially when the maximal number of calls from a category increases, or the maximal queue size increases. In this paper, we have investigated two approximations which reduce the MDP computational complexity to manegeable levels, namely state aggregation and decomposition of the link Markov process.

The exact and approximate link MDP models were tested in an extensive numerical study. The results show that the MDP-based methods acheives the maximum value of the objective function, in contrast to the LLP routing method. The approximate MDP method based on link

Markov process decomposition shows good performance under delayed call set up. However, under pure on demand call set up, the same method performs poorly in the studied simulation example. Fortunately, in this case, the state aggregation method acheives near-optimal performance.

References

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

[2] 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,*, pp. 39-46, IEEE Computer Society Press, 1990.

[3] Dziong Z., Mason L., "Call Admission and Routing in Multi-Service Loss Networks", *IEEE Trans. on Commun.*, Vol. 42, No. 2, 1994

[4] Dziong Z., Nordström E., "CAC and Routing for Multi-Service Networks with Delayed Set Up of Wide-band Calls – Markov Decision Theory Framework", submitted, 2002.

[5] Liao K.-Q., Mason L.G., "An Approximate Performance Model for a Multislot Integrated Services System", IEEE Trans. Commun., vol.37, pp. 211- 221, March 1989.

[6] Kaufman J., "Blocking in a Shared Resource Environment", *IEEE Transactions on communications*, Vol. COM-29, No. 10, pp. 1474-1481, 1981.

[7] Krishnan K., Hübner S., "Admission Control and Routing for Multirate Circuit-Switched Traffic",ITC 15, Washington, 1997.

[8] Nordström E., "Near Optimal Link Allocation of Blockable Narrow-band and Queueable Wide-Band Call Traffic in ATM Networks",ITC 15, Washington, 1997.

[9] Nordström E., Carlström J., "Call Admission Control and Routing in Multi-Service Networks by Markov Decision Theory with State Aggregation", submitted, 2002.

[10] Roberts J., “A Service System with Heterogeneous User Requirements”, *Performance of Data Communications Systems and Their Applications*, Editor G. Pujolle, North-Holland, 1981.

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