EUROPEAN TRANSACTIONS ON TELECOMMUNICATIONS
Euro. Trans. Telecomms. 2006; **17**:21–36
Published online 5 July 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/ett.1056

Communication Networks

CAC and routing for multi-service networks with blocked wide-band calls delayed, part I: exact link MDP framework

Ernst Nordström[1]* and Zbigniew Dziong[2]

[1]*Department of Culture/Media/Computer Science, Dalarna University, SE-781 88 Borlänge, Sweden*
[2]*École de technologie supérieure, Montréal, Québec, Canada*

SUMMARY

In this paper, we study the call admission control (CAC) and routing issue in multi-service networks. Two categories of calls are considered: a narrow-band (NB) with blocked calls cleared and a wide-band (WB) with blocked calls delayed. The objective function is formulated as reward maximisation with penalty for delay. The optimisation is subject to quality of service (QoS) constraints and, possibly, grade of service (GoS) constraints. A suboptimal solution is achieved 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 Markov and reward processes respectively. Second, the dimensions of the link Markov and reward processes are reduced by aggregation of the call classes into call categories. The CAC and routing policy is computed by the policy iteration algorithm from MDP theory. The numerical results show that the proposed CAC and routing method, based on the exact link MDP framework, is able to find an efficient trade-off between reward loss and average call set-up delay, outperforming conventional methods such as the least loaded routing (LLR). Copyright © 2005 AEIT.

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 maximise the revenue from carried calls, while meeting constraints on the quality of service (QoS) and grade of service (GoS) on the packet and call level respectively.

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

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 [1]. There exist two main approaches to cope with this fairness problem: access control of NB calls or queuing of WB calls.

Trunk reservation is a form of access control which reserves capacity to WB calls by rejecting NB calls when the link occupancy is over a threshold. While access

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

Contract/grant sponsor: Swedish Knowledge Foundation.

Received 23 August 2004
Revised 17 November 2004
Accepted 15 February 2005

control can deliver good fairness properties, this is usually achieved at the expense of bandwidth utilisation.

Queuing of a WB call request is done when the network is found busy by the WB call request. When a sufficient amount of bandwidth becomes available in the network, a waiting WB call is allowed to enter the network. This approach, if applied correctly, can provide access fairness and increased bandwidth utilisation at the same time.

Guérin [2] studied a single link cutoff priority system for two Poisson call arrival streams with homogeneous bandwidth requirements, distinct arrival rates and common exponential service time distribution. Calls of type I have access to the link with no restriction, but will be blocked if all servers are busy. Calls of type II have free access to the servers as long as there are more than g servers free. In case only g or fewer server remains free, calls of type II will be queued and will receive service on a first come, first served basis as soon as more than g servers become idle. The authors used z-transformation techniques to compute system performance; that is, the mean waiting time in the queue and the probability of delay for type II calls, as well as the probability of blocking for type I calls.

Serres and Mason [1] considered a single link cutoff priority system for two Poisson call arrival streams with heterogeneous bandwidth requirements, distinct arrival rates and exponential service time distributions. Calls of type I have access to the link with no restriction, but will be blocked if all servers are busy. Calls of type II have restricted access to the service facility; the cutoff parameter r_0 specifies the maximum number of type II calls that can be in service at the same time. Therefore, a queue of type II calls forms as soon as a type II arrival finds r_0 type II calls already in service or otherwise if there are not enough free servers. The authors proposed two ways for obtaining the system performance, namely moment-generating functions and matrix-geometric techniques.

Modern CAC and routing mechanisms are state-dependent rather than static, which means that the decision to reject the request for a new call, or to accept it on a particular path depends on the current occupancy of the network. That is, the state of the network is represented by the number of calls from each class in service, or waiting for service, at each network link. A state-dependent CAC and routing policy is a mapping, for every call class, from a network state space to a set of possible routing decisions, see Figure 1. First, CAC determines the set of feasible paths between the source and destination which offers sufficient QoS to the new and existing calls in terms of delay, delay variation and data loss. Second, the network should select one path among the set of feasible paths to convey the call or to reject it if call acceptance would diminish the expected revenue. While contributing to the maximisation of the average revenue for the operator, this choice must comply with GoS constraints in terms of call blocking probabilities and call set-up delays. State-dependent mechanisms offer advantages both in terms of achievable revenue and ability to control the QoS and GoS.

This paper deals with a particular form of state-dependent CAC and routing, where the behaviour of the network is formulated as Markov decision process (MDP) [3, 4]. An MDP is a controlled Markov process, where the set of state transitions from the current Markov state to other

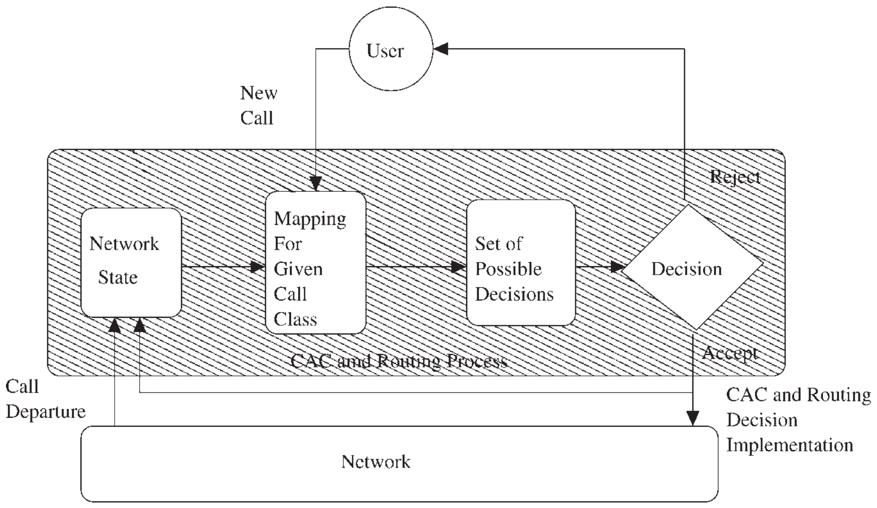
User
New Call
Network State
Mapping For Given Call Class
Set of Possible Decisions
Decision
Reject
Accept
CAC and Routing Process
Call Departure
CAC and Routing Decision Implementation
Network

Figure 1. State-dependent CAC and routing.

Markov states depends on the decision or action taken by the controller in the current state. Reward delivery from the user to the network can be modelled as occurring at call completion since this provides a correct model of carried reward.

The computational burden of the exact MDP framework for CAC and routing is prohibitive even for moderate-size networks. Fortunately, it can be reduced to manageable levels by a set of modelling simplifications. First, the network is decomposed into a set of links assumed to have independent traffic and reward processes respectively. Second, the K dimensional link Markov process and link reward process are aggregated into G dimensional link Markov process and link reward processes respectively. Third, as will be studied in part II of this paper [Nordström E, Dziong Z. CAC and routing in multi-service networks with blocked wide-band calls delayed, part II: approximate link MDP framework, submitted for publication], the G dimensional link Markov process and link reward process can be decomposed into per-category link Markov processes and link reward processes.

Dziong *et al*. proposed in Reference [5] an MDP framework for CAC and routing with blocked NB calls cleared and blocked WB calls delayed. The control objective was formulated as minimisation of a cost function being a linear combination of the reward loss due to rejection of NB calls and the average call set-up delay of WB calls. The average delay was multiplied by a weight which provides a tool for controlling the trade-off between NB reward losses and average WB call set-up delay. The numerical results showed that routing based on the MDP framework achieved a lower value of the objective cost function than the routing based on load sharing.

In this paper, we propose a new MDP framework for CAC and routing with blocked NB calls cleared and blocked WB calls delayed. In this framework, we describe the network process as a reward process with the objective of reward maximisation (which is the inverse of cost minimisation). Then we formulate our new control objective by defining the reward as a linear combination of the reward from accepted NB and WB calls, and the average call set-up delay of WB calls. Hence, we consider the reward contribution from both NB and WB calls, and not just from the NB calls as was the case in Reference [5]. Since both NB and WB calls are accounted for in the control objective it becomes possible to control the access fairness (call blocking probability) among both the NB and WB classes. The trade-off between NB and WB reward loss and average WB call set-up delay is controlled by the weight of the average delay term.

The contribution of this paper is twofold. First, we present an MDP-based solution to a new formulation of CAC and routing in mixed loss-delay call set-up mode. The objective function is formulated as reward maximisation with penalty for average call set-up delay. The optimisation is subject to QoS constraints and possibly GoS constraints. Second, we present an extensive numerical evaluation, based on simulation, of the MDP-based method for CAC and routing. For comparison, the performance of the least loaded routing (LLR) algorithm is also evaluated.

The paper is organised as follows. Section 2 formulates the CAC and routing problem in terms of offered traffic, network and queuing model, QoS and GoS constraints and optimisation objective. Section 3 describes the network model and the exact link MDP model. Section 4 outlines the MDP computation procedure. Section 5 gives a summary of the numerical/simulation-based evaluation of the performance of the MDP-based method as well as the LLR method. Finally, Section 6 concludes the paper.

2. PROBLEM FORMULATION

2.1. Traffic assumptions

The network is offered traffic from K classes which are, for sake of simplicity, subject to deterministic multiplexing. The j-th class, $j \in J = \{1, \ldots, K\}$, is characterised 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
- Reward parameter $r_j \in (0, \infty)$

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

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

- Bandwidth requirement $b_i \in \{b_\mathrm{n}, b_\mathrm{w}\}$ [Mbps]

Table 1. Symbols used in the paper (a).

Network CAC and routing policy	π
Link CAC policy	π^s
Number of OD pairs	P
Number of call classes	K
Number of call categories	G
Link index	s
Path index	k
Class index	j
Category index	i
Set of link indices	S
Set of link indices for path k	S_k
Set of class indices	J
Set of category indices	I
Bandwidth requirement	b_j
Call arrival rate	λ_j
Call holding time	$1/\mu_j$
Set of alternative routes	W_j
Number of alternative routes	H
Reward parameter	r_j
Average reward parameter	$\bar{r}_i^s\pi$
Average mean holding time	$1/\bar{\mu}_i^s$
Offered network reward	R
Carried objective reward	$\bar{R}_D$
Carried network reward	$\bar{R}$
Average call set-up delay	$\bar{D}$
Power ratio	PR
Fuzzy reward measure	S_{R}
Fuzzy delay measure	S_{D}
Delay penalty weight	α

Table 2. Symbols used in the paper (b).

State space	X
Network state	$\mathbf{z}$
Link state	$\mathbf{x}, \mathbf{y}$
Number of class j calls	z_j
Queue state	z_l^s, x_l^s
Link capacity	C^s
Queue capacity	L^s
Action space	A
Action vector	$\mathbf{a}$
State transition probability	$p_{\mathbf{xy}}(\mathbf{a})$
Link reward parameter	$r_j^s(\pi)$
Rate of offered calls	$\lambda_j^k(\pi)$
Rate of accepted calls	$\bar{\lambda}_j^s$
Link call arrival rate	$\lambda_i^s(\mathbf{x}, \pi)$
Blocking probability	$B_j^c(\pi)$
Filtering probability	$\phi_{jk}^s(\mathbf{x}, \pi)$
Category probability	p_{ij}^s
Average sojourn time	$\tau(\mathbf{x}, \mathbf{a})$
Expected reward	$R_D^s(\mathbf{x}, \mathbf{a})$
Reward rate	$q^s(\mathbf{x})$
Path net-gain	$g_j^k(\mathbf{y}, \pi)$
Link net-gain	$g_i^s(\mathbf{x}, \pi)$
Link shadow price	$p_i^s(\mathbf{x}, \pi)$
Path shadow price weight	β
Increment vector	δ_i
Relative value	$v^s(\mathbf{x}, \pi)$
Reference state	$\mathbf{x}_r$
Trunk reservation parameter	θ_j^n
Total offered traffic load	ρ

- Average mean call holding time $1/\bar{\mu}_i^s$ [s]
- Average reward parameter $\bar{r}_i^s(\pi)$

where π denotes the CAC and routing policy.

The symbols used in this paper are listed in Tables 1–3.

2.2. Network and queuing model

The network is assumed to consist of a set of switching nodes. The switching nodes communicate in both traffic flow directions using uni-directional links. Each uni-directional link has one finite FIFO queue for WB call requests. One can consider two basic schemes of queuing system management. In the first the queues operates as follows [5]: 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 OD node. Otherwise, at least one link along the path is not able to directly accept the new WB call. At those links, the new WB call request joins the queue at the tail. We assume that the path would not be chosen when some of its links have insufficient capacity on both the link and in the queue. At links with sufficient capacity, bandwidth is reserved for the new WB call while waiting for all links to be ready to accept the call. A link queue is served when a sufficient number or bandwidth units become 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 OD node.

The advantage of this scheme is its simplicity but its performance may have some drawbacks. One is the 'reservation' traffic caused by multi-link calls due to bandwidth reservation on some links while the call request is in the queue of other links. Although this 'reservation' traffic is

Table 3. Symbols used in the paper (c).

#simulation points per curve	N
#simulation runs per point	M
Pooled variance	s^2
Variance of pooled variance	S^2
Reward loss	L
Objective reward loss	L_D

likely to be negligible under nominal conditions, it can be significant in case of overloads.

In the second scheme of queuing system management, each link has two FIFO queues. One of these queues has non-preemptive priority. In the case of one-link path, the connection request is placed in the non-priority queues. In the case of multi-link path, the call request is placed in the non-priority queue of the first link. When the request is at the head of the queue and the priority queue is empty, the requested bandwidth is reserved (when available) and the call request is placed in the priority queues of all other links belonging to the path. This approach provides that the 'reservation' traffic is reduced significantly, compared to the first scheme, and the average waiting time for one-link connections is similar to the one for multi-link connections (at least in nominal traffic conditions). These advantages are balanced by increased complexity of queuing system management.

In this paper, we consider queue management corresponding to the first scheme. The advantage of this approach is that analytical models are relatively simple. The performance drawbacks of this scheme are not affecting our study objective as we are trying to analyse features of different options of CAC and routing algorithms, which are on the higher level of the control hierarchy. In other words, all of the compared control strategies use the same queuing management scheme and therefore we believe that the qualitative conclusions of our study do not depend strongly on the chosen queue management scheme.

2.3. QoS and GoS constraints

CAC and routing faces QoS constraints and, possibly, GoS constraints. First, the $\mathrm{CAC}_{\mathrm{QoS}}$ function finds the set of feasible paths that comply with the end-to-end QoS constraints of the requested call class. Second, the routing function selects a path for the new call. Third, the $\mathrm{CAC}_{\mathrm{GoS}}$ function accepts/rejects this choice based on revenue considerations and end-to-end GoS constraints of the requested call class.

The QoS measures include packet delay, packet delay variation and packet loss probability. To simplify CAC, each link typically has a target QoS level that it should maintain. The link QoS constraint is used by the MDP routing controller to determine the set of feasible link states.

The GoS measures include call blocking probability and the call set-up delay. As already mentioned, the reward parameters offers a flexible tool for controlling the GoS among the call classes. The reward parameters that are in agreement with the revenue objective of the network operator and the GoS demands of the users can be determined by some automated search procedure. The end-to-end call blocking probabilities for a given configuration of reward parameters can be determined from a set of fixed point equations [6].

2.4. Objective function

In our loss-delay type of system, we have to deal with bi-objective type function for the CAC and routing policy π. Let us first define each objective separately. To take into account traffic losses due to the 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 reward parameter and departure rate of the j-th type connection respectively. Now we can define the objective function, $\bar{R}$, as average reward from the network given by

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

where $\bar{\lambda}_j$ denotes the j-th class connection acceptance rate (the process is assumed to be stationary). The obvious goal of the CAC and routing algorithm is to maximise the objective function. This approach was already applied in state-dependent routing schemes for loss systems presented in References [7, 8]. The reward maximisation 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. [7, 8]). Another advantage is that by using the control model presented in References [9, 10], the objective function can be decomposed as follows

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

where $\bar{R}^s$ 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:

$$\bar{R} = \bar{R}_{\mathrm{n}} + \bar{R}_{\mathrm{w}} = \sum_s [\bar{R}^s_{\mathrm{n}} + \bar{R}^s_{\mathrm{w}}] \tag{3}$$

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

$$\bar{D} = \sum_s \bar{D}^s \frac{\lambda^s_{\mathrm{w}}}{\lambda_{\mathrm{w}}} \tag{4}$$

where $\bar{D}^s$ denotes the average delay of calls in the s-th link queue and λ_{w}^s, λ_{w} denotes the arrival rate of WB calls offered to the s-th link and totally, to all OD pairs of the network respectively.

In general the presented two objectives are conflicting, that is when $\bar{D}$ is increased by the control, $\bar{R}$ is also increased and *vice versa*. Thus the global objective function must provide a mechanism to trade-off the mentioned two objectives. Such functions, for loss-delay systems, were presented in References [9, 10]. In Reference [9], the power factor, PR, was used to evaluate link performance. Using this concept for the problem under consideration the objective would be to maximise the following expression

$$\mathrm{PR} = \frac{\bar{R}/R}{1+\bar{D}} \tag{5}$$

where R is the offered reward to the network. In Reference [10], the objective function was based on the fuzzy-set approach (see Reference [11]). In our context it would be maximisation of the following function

$$F = \min(S_{\mathrm{R}}(\bar{R}), S_{\mathrm{D}}(\bar{D})) \tag{6}$$

where the functions $S_{\mathrm{R}}, S_{\mathrm{D}}$ are subjective measures of 'controller satisfaction' with the reward and delay performance respectively. For example, non-linear functions $S_{\mathrm{R}}(\bar{R}), S_{\mathrm{D}}(\bar{D})$ can transform the GoS measures $\bar{R}, \bar{D}$ into the values from [0,1] interval where 0 denotes good performance and 1 denotes unacceptable performance.

Although both approaches have in general some desirable features, there is one important drawback from our problem point of view. Namely, due to the non-linear nature of these formulations the resulting global objective functions are not decomposable into a set of separable link objective functions and, as it will become obvious later on, this feature is critical for achieving a practical solution. For this reason we select for the objective function a linear combination of the two, which can also be interpreted as reward maximisation with penalty for delay of WB calls:

$$\bar{R}_D = \bar{R} - \alpha\bar{D} \tag{7}$$

where α is the delay penalty weight which determines the trade-off value between the reward and average delay changes. By using Equations (2), (4) in Equation (7) we arrive at

$$\bar{R}_D = \sum_s \left[\bar{R}^s - \alpha\bar{D}^s \frac{\lambda_{\mathrm{w}}^s}{\lambda_{\mathrm{w}}}\right] \tag{8}$$

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

Note that the form of Equation (8) suggests that the delay penalty weight can also be link dependent:

$$\bar{R}_D = \sum_s \left[\bar{R}^s - \alpha^s\bar{D}^s \frac{\lambda_{\mathrm{w}}^s}{\lambda_{\mathrm{w}}}\right] \tag{9}$$

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

The behaviour of the network under consideration can be described by an MDP with the objective to maximise the reward function defined by Equation (9). The corresponding reward rate, $q(\mathbf{z})$, is given by:

$$q(\mathbf{z}) = \sum_{j\in J_{\mathrm{n}}} r_j z_j \mu_{\mathrm{n}} + \sum_{j\in J_{\mathrm{w}}} r_j z_j \mu_{\mathrm{w}} - \sum_{s\in S} \alpha^s \frac{z_l^s}{\lambda_{\mathrm{w}}} \tag{10}$$

where $\mathbf{z}$ denotes the network state and z_j, z_l^s denote the number of the j-th type calls and the number of calls in the s-th queue in state $\mathbf{z}$ respectively, J_{n} denotes the set of NB classes, J_{w} denotes the set of WB classes, and S denotes the set of all link indices in the network.

The action space is given by

$$A = \{a = \{a_j\} : a_j \in \{0\} \cup W_j, j \in J\} \tag{11}$$

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

The network state and action spaces can be very large, even for moderate-sized networks. We therefore decompose the network into a set of links assumed to have independent traffic and reward processes respectively [12].

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 consisting of l links is decomposed into l independent link calls characterised 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 $r_j^s(\pi)$ fulfil the obvious condition that

$$r_j = \sum_{s\in S_k} r_j^s(\pi) \tag{12}$$

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 [12]. In this paper, we use a simple rule: the call reward is distributed uniformly among the path's links, resulting in the formula $r_j^s(\pi) = r_j/l$, where l denotes the number links in the call's path.

Even in the decomposed network model, the state space can be quite large when many call classes share the links. One way to reduce the state space is to construct a modified link reward process in which the link call *classes* with the same bandwidth requirement are aggregated into one *category* $i \in I$ with average reward parameter defined as [12]:

$$\bar{r}_i^s(\pi) = \frac{\sum_{j\in J_i} r_j^s(\pi)\bar{\lambda}_j^s(\pi)}{\sum_{j\in J_i} \bar{\lambda}_j^s(\pi)} \tag{13}$$

where J_i denotes the set of classes that belongs to the i-th category, and $\bar{\lambda}_j^s(\pi)$ 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 presents the exact link MDP model. The state in the exact link model is given by $\mathbf{x} = (x_\mathrm{n}, x'_\mathrm{w})$, 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:

$$X = \{\mathbf{x} = (x_\mathrm{n}, x'_\mathrm{w}) : 0 \leqslant x_\mathrm{n} \leqslant N_\mathrm{n}^s, 0 \leqslant x'_\mathrm{w} \leqslant N_\mathrm{w}^s + L^s, x_l = f_l(\mathbf{x}), x_\mathrm{n}b_\mathrm{n} + (x'_\mathrm{w} - x_l)b_\mathrm{w} \leqslant C^s\} \tag{14}$$

where $N_\mathrm{n}^s = \lfloor C^s/b_\mathrm{n}\rfloor$, $N_\mathrm{w}^s = \lfloor C^s/b_\mathrm{w}\rfloor$ 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:

$$x_l = f_l(\mathbf{x}) := \inf\{x_l : x_l \geqslant 0, C^s - x_\mathrm{n}b_\mathrm{n} \geqslant (x'_\mathrm{w} - x_l)b_\mathrm{w}\} \tag{15}$$

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

$$A = \{\mathbf{a} = \{a_i\} : a_i \in \{0,1\}, i \in I\} \tag{16}$$

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:

$$A(\mathbf{x}) = \{\mathbf{a} \in A : a_i = 0 \text{ if } \mathbf{x} + \delta_i \notin X, i \in I\} \tag{17}$$

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

The Markov chain is characterised by state transition probabilities $p_{\mathbf{xy}}(\mathbf{a})$ which express the probability that the next state is $\mathbf{y}$, given that action $\mathbf{a}$ is taken in state $\mathbf{x}$. The transition probabilities fulfil the obvious constraint $\sum_{\mathbf{y}\in X} p_{\mathbf{xy}}(\mathbf{a}) = 1$. In our case, the state transition probabilities become:

$$p_{\mathbf{xy}}(\mathbf{a}) = \begin{cases} \lambda_i^s(\mathbf{x},\pi)a_i\tau(\mathbf{x},\mathbf{a}), & \mathbf{y} = \mathbf{x} + \delta_i \in X, i \in I \\ x_\mathrm{n}\bar{\mu}_\mathrm{n}^s\tau(\mathbf{x},\mathbf{a}), & \mathbf{y} = \mathbf{x} - \delta_\mathrm{n} \in X \\ (x'_\mathrm{w} - x_l)\bar{\mu}_\mathrm{w}^s\tau(\mathbf{x},\mathbf{a}), & \mathbf{y} = \mathbf{x} - \delta_\mathrm{w} \in X \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

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 $\bar{\mu}_\mathrm{n}^s$, $\bar{\mu}_\mathrm{w}^s$ denotes the departure rate of NB and WB calls respectively, and $\tau(\mathbf{x},\mathbf{a})$ denotes the average sojourn time in state $\mathbf{x}$. The link call arrival rates, $\lambda_i^s(\mathbf{x},\pi)$, are given by:

$$\lambda_i^s(\mathbf{x},\pi) = \sum_{j\in J_i}\sum_{k\in W_j^s} \lambda_j^k(\pi)\phi_{jk}^s(\mathbf{x},\pi) \prod_{c\in S_k\setminus\{s\}} (1 - B_j^c(\pi)) \tag{19}$$

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, W_j^s denotes the set of alternative paths for class j that traverses link s and $\phi_{jk}^s(\mathbf{x},\pi)$ denotes a filtering probability defined as:

$$\phi_{jk}^s(\mathbf{x},\pi) = P\left\{\sum_{c\in S_k\setminus\{s\}} p_j^c(\mathbf{x},\pi) < r_j - p_j^s(\mathbf{x},\pi) \mid \bar{B}_j\right\} \tag{20}$$

where $\bar{B}_j$ denotes the condition that no link on path k is in the blocking state, $p_j^c(\mathbf{x},\pi)$ denotes the state-dependent *link shadow price* for class j on link c (note that $p_j^s(\mathbf{x},\pi)$ is constant in Equation (21)). 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 [6], or approximated with one according to experiments in Reference [12]. The $\lambda_j^k(\pi)$ denotes the arrival rate of class j to path $k \in W_j$, and is given by the following load sharing model [12]:

$$\lambda_j^k(\pi) = \lambda_j \frac{\bar{\lambda}_j^k(\pi)}{\sum_{h\in W_j} \bar{\lambda}_j^h(\pi)} \tag{21}$$

where the $\bar{\lambda}_j^k(\pi)$ 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:

$$\bar{\mu}_{\mathrm{n}}^{s} = \left[\sum_{j \in J_{\mathrm{n}}} p_{\mathrm{n}j}^{s} \mu_{j}^{-1} \right]^{-1} \tag{22}$$

where $p_{\mathrm{n}j}^{s}$ denotes the probability that an arbitrary NB call found on the link is from class $j \in J_{\mathrm{n}}$:

$$p_{\mathrm{n}j}^{s} = \frac{\bar{\lambda}_{j}^{s}(\pi)}{\sum_{c \in J_{\mathrm{n}}} \bar{\lambda}_{c}^{s}(\pi)} \tag{23}$$

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

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

$$\bar{\mu}_{\mathrm{w}}^{s} = \left[\sum_{j \in J_{\mathrm{w}}} p_{\mathrm{w}j}^{s} \mu_{j}^{-1} + T^{s} \right]^{-1} \tag{24}$$

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

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

$$\tau(\mathbf{x}, \mathbf{a}) = \left\{ \sum_{i \in I} x_i \bar{\mu}_{i}^{s} + a_i \lambda_{i}^{s}(\mathbf{x}, \pi) \right\}^{-1} \tag{25}$$

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

$$q^{s}(\mathbf{x}) = \bar{r}_{n}^{s}(\pi) x_n \bar{\mu}_{n}^{s} + \bar{r}_{w}^{s}(\pi)(x' - x_1)\bar{\mu})_{w}^{s} - \alpha^{s} \frac{x_1}{\lambda_w} \tag{26}$$

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 offers maximal positive path net-gain among the paths with sufficient QoS (See Figure 2). The call is rejected if the maximum path net-gain is negative, or if no path would offer sufficient QoS.

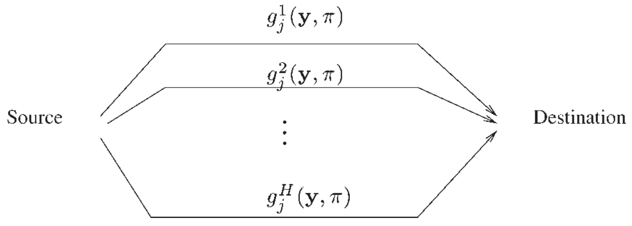

Figure 2. The call is offered to a path which has sufficient QoS and maximal positive path net-gain among the $H = |W_j|$ alternative paths.

4.1. Basic definitions

The link independence assumption can result in overestimation of the path net-gain for multi-link paths [5]. Obviously, real networks have certain positive correlation between states on different links. For this reason, the following definition of the path net-gain has been proposed by Dziong and Mason [12]. The path net-gain is obtained as a weighted sum of state-dependent and average path net-gain:

$$\begin{aligned} g_{j}^{k}(\mathbf{y}, \pi) &= (1-\beta) g_{j}^{k}(\mathbf{y}, \pi) + \beta E_c[g_{j}^{k}(\mathbf{y}, \pi)] \\ &= r_j - \sum_{s \in S_k} [(1-\beta) p_{i}^{s}(\mathbf{x}) + \beta \bar{p}_{i}^{s}(\pi)] \end{aligned} \tag{27}$$

where $\beta \in [0,1]$ is a weighting factor, $E_c[\ldots]$ denotes the expectation over all states occurring at call arrivals, i is the category of class j and $p_{i}^{s}(\mathbf{x}, \pi)$ denotes the state-dependent link shadow price for category i, and $\bar{p}_{i}^{s}(\pi)$ denotes the average link shadow price for category i. The average link shadow price is obtained from the state-dependent link shadow prices as follows:

$$\bar{p}_{i}^{s}(\pi) = E_c[p_{i}^{s}(\mathbf{x}, \pi)] = \sum_{\mathbf{x} \in X} Q_i(\mathbf{x}) p_{i}^{s}(\mathbf{x}, \pi) \tag{28}$$

where $Q_i(\mathbf{x})$ denotes the probability that the i-th category call is accepted in state $\mathbf{x}$. Note that $\bar{p}_{i}^{s}(\pi)$ can be estimated in a real network by averaging the values of link shadow price at the instants of the i-th type call arrivals.

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

$$g_{j}^{k}(\mathbf{y}, \pi) = r_j - \sum_{s \in S_k} p_{i}^{s}(\mathbf{x}, \pi) \tag{29}$$

where $\mathbf{y} = \{\mathbf{x}\}$ denotes the network state in the exact network model and i is the category of class j. The link shadow price $p_{i}^{s}(\mathbf{x}, \pi)$ can be interpreted as the expected cost for accepting an i-th category call in state $\mathbf{x} = (x_{\mathrm{n}}, x'_{\mathrm{w}})$ and is defined as follows:

$$p_{i}^{s}(\mathbf{x}, \pi) = \bar{r}_{i}^{s}(\pi) - g_{i}^{s}(\mathbf{x}, \pi) \tag{30}$$

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 exact link models, as follows:

$$g_i^s(\mathbf{x}, \pi) = v^s(\mathbf{x} + \delta_i, \pi) - v^s(\mathbf{x}, \pi) \tag{31}$$

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, $R_D^s(\mathbf{x}_0, \pi, T)$, obtained in a interval $(t_0, t_0 + T)$ of length T, assuming state $\mathbf{x}_0$ at time t_0:

$$R_D^s(\mathbf{x}_0, \pi, T) = E\left[\int_{t_0}^{t_0+T} q^s(\mathbf{x}(t))\,\mathrm{d}t\right] \tag{32}$$

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:

$$v^s(\mathbf{x}_0, \pi) = \lim_{T\to\infty}\left[R_D^s(\mathbf{x}_0, \pi, T) - R_D^s(\mathbf{x}_\mathrm{r}, \pi, T)\right] \tag{33}$$

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, $\mathbf{x}_\mathrm{r}$. 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 summarised as follows:

1. **Startup**: Initialise the relative values in a way that makes all link net-gains with permissible admission positive.
2. **On-line operation phase**: Measure per-path call acceptance rates $\bar{\lambda}_j^k(\pi)$ 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 $\bar{r}_i^s(\pi)$ 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 $\bar{R}_D^s(\pi)$ for the current 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 point 2 until average reward per time unit converges.

According to an MDP theory an optimal policy is found after a finite number of policy iterations in case of a finite state and policy space [4].

4.2.1. Value determination for the exact link model. The *value determination step* for link s for the exact link model determines the relative values $v^s(\mathbf{x}, \pi)$ for all states $\mathbf{x} \in X$ and the average reward rate $\bar{R}_D^s(\pi)$ by solving a sparse system of linear equations:

$$\begin{cases} v^s(\mathbf{x}, \pi) = R_D^s(\mathbf{x}, \mathbf{a}) - \bar{R}_D^s(\pi)\tau(\mathbf{x}, \mathbf{a}) + \sum_{\mathbf{y}\in X} p_{\mathbf{xy}}(\mathbf{a}) v^s(\mathbf{y}, \pi) \\ v^s(\mathbf{x}_\mathrm{r}, \pi) = 0; \quad \mathbf{x} \in X \backslash \{\mathbf{x}_r\} \end{cases} \tag{34}$$

where the following quantities need to be specified:

- X: the state space, that is the set of possible states,
- $\mathbf{a} = \pi^s(\mathbf{x})$: the control action in state $\mathbf{x}$,
- $\tau(\mathbf{x}, \mathbf{a})$: the expected sojourn time in state $\mathbf{x}$,
- $R_D^s(\mathbf{x}, \mathbf{a})$: the expected link reward when leaving state $\mathbf{x}$,
- $p_{\mathbf{xy}}(\mathbf{a})$: the transition probability from state $\mathbf{x}$ to state state $\mathbf{y}$, given that action $\mathbf{a}$ is taken in state $\mathbf{x}$ and
- $\mathbf{x}_\mathrm{r}$: 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
- $\bar{R}_D^s(\pi)$: 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.

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

$$\mathbf{a} = \max_{\mathbf{u}\in A(\mathbf{x})}\left\{R_D^s(\mathbf{x}, \mathbf{u}) - \bar{R}_D^s(\pi)\tau(\mathbf{x}, \mathbf{u}) + \sum_{\mathbf{y}\in X} p_{\mathbf{xy}}(\mathbf{u}) v^s(\mathbf{y}, \pi)\right\} \tag{35}$$

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. Three state-dependent MDP-based routing algorithms are compared:

- MDP—standard MDP routing based on exact link model proposed in Subsection 3.2,
- MDP_P—MDP routing based on the exact link model with priority for the shortest path,
- MDP_W—MDP routing based on the exact link model with path net-gain obtained as a weighted sum of state-dependent and average path net-gain according to Equation (27).

The presence of multi-link calls will introduce positive correlation between states on successive links. This correlation is not modelled by the standard MDP routing method, which is based on the link independence assumption and the exact link model. Hence, the MDP_P and MDP_W routing methods try to compensate for the error caused by the link independence assumption.

The performance of a popular conventional routing method, namely LLR is also evaluated. The reason for choosing LLR is that it is with best performance among the routing methods [13, 8]. 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. In case of routing of NB calls, a path among this set 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. In case of routing of WB calls, the paths which require no queuing are given priority. The same routing rule as for NB calls applies to the set of shortest paths, which requires no queuing. In case queuing is necessary, the WB call is offered to the path which has the smallest maximal queue length along the path. For both NB and WB calls, if all paths are busy, 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 link/queue capacity.

The priority mechanism in MDP_P routing works as follows. The set of shortest paths with sufficient free link/queue 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 all paths between the OD pair are busy.

5.2. Examples and results

The performance analysis is performed for the network example W6N described in Table 4. The total offered traffic load is measured by $\rho = \sum_{j \in J} b_j \lambda_j \mu_j^{-1}$ [Mbps*Erlang]. The topology of W6N is shown in Figure 3. The link capacities and offered traffic volumes for network example W6N are based on the example in Reference [14] and is shown in Table 5. Network W6N corresponds to an STM type network. The OD pairs in W6N are offered different traffic volumes (asymmetric case). The algorithm-specific parameter settings, presented in Table 6, were determined heuristically based on simulation experience.

Table 4. Description of network example W6N.

	W6N
Symmetrical	No
#nodes	6
#uni-directional links	30
#OD pairs P	30
#routes per OD pair	5
Link capacity C^s [Mbps]	12–192
Queue capacity L^s	0–3
Network capacity [Mbps]	2484
Max #links in path	2
#traffic categories G	2
Mean holding time $1/\mu_j$ [s]	1, 10
Bandwidth b_j [Mbps]	1, 6
Total offered load ρ [Mbps*Erlang]	1816.8
$r'_j = r_j \mu_j / b_j$	1

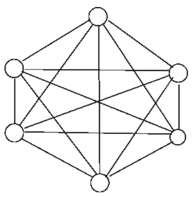

Figure 3. Network example W6N.

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 loss variance, and the variance of the pooled loss variance, in Table 7–11. We compute the pooled variance over the N simulation points as:

$$s^2 = \frac{1}{N}\sum_{k=1}^{N} s_k^2 \tag{36}$$

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

$$s_k^2 = \frac{1}{M-1}\sum_{i=1}^{M}(x_{ik} - \bar{x}_k)^2 \tag{37}$$

The sample variance of the pooled variance is obtained as:

$$S^2 = \frac{1}{N-1}\sum_{k=1}^{N}(s_k^2 - s^2)^2 \tag{38}$$

Table 5. 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 6. Algorithm specific parameters.

MDP adaptation epochs	6
MDP_P adaptation epochs	6
MDP_W adaptation epochs	6
Call events in warm up period	500 000
Call events in adaptation period	1 000 000
Call events in measurement period	1 000 000
Delay penalty weight α^s	100
MDP_W weight β in Equation (28)	0.5
#simulation points per curve N	4, 16, 19
#simulation runs per point M	20
Trunk reservation parameter θ_j^n	0
Filtering probability $\phi_j^s(\mathbf{x}, \pi)$	1.0

Table 7. Pooled variance in simulations with variable delay penalty weight.

	L [%] $s^2(S^2)$	$\bar{D}$ [s] $s^2(S^2)$	L_D [%] $s^2(S^2)$
MDP	0.05 (0.0006)	0.00 (0.0000)	0.08 (0.0013)
MDP_P	0.04 (0.0002)	0.00 (0.0000)	0.06 (0.0003)
MDP_W	0.04 (0.0003)	0.00 (0.0000)	0.10 (0.0014)
LLR	0.04 (0.0003)	0.00 (0.0000)	0.17 (0.0184)

Table 8. Pooled variance in simulations with variable maximal queue size.

	L [%] $s^2(S^2)$	$\bar{D}$ [s] $s^2(S^2)$	L_D [%] $s^2(S^2)$
MDP	0.07 (0.0011)	0.00 (0.0000)	0.09 (0.0009)
MDP_P	0.04 (0.0009)	0.00 (0.0000)	0.05 (0.0007)
MDP_W	0.06 (0.0020)	0.00 (0.0000)	0.10 (0.0028)
LLR	0.06 (0.0020)	0.00 (0.0000)	0.08 (0.0014)

Table 9. Pooled variance in simulations with variable traffic ratio for $L^s = 0$ case.

	L [%] $s^2(S^2)$
MDP	0.16 (0.0053)
MDP_P	0.10 (0.0057)
MDP_W	0.17 (0.0045)
LLR	0.15 (0.0154)

Table 10. Pooled variance in simulations with variable traffic ratio for $L^s = 3$ case.

	L [%] $s^2(S^2)$	$\bar{D}$ [s] $s^2(S^2)$	L_D [%] $s^2(S^2)$
MDP	0.03 (0.0004)	0.00 (0.0000)	0.05 (0.0008)
MDP_P	0.02 (0.0001)	0.00 (0.0000)	0.03 (0.0005)
MDP_W	0.05 (0.0070)	0.00 (0.0000)	0.09 (0.0174)
LLR	0.03 (0.0004)	0.00 (0.0000)	0.07 (0.0026)

Table 11. Pooled variance in simulations with variable WB normalised reward parameter.

	NB blocking probability [%] $s^2(S^2)$	WB blocking probability [%] $s^2(S^2)$
MDP	58.3 (5738.4)	1.3 (7.2)
MDP_P	6.4 (122.3)	1.65 (19.5)
MDP_W	51.0 (4593.4)	2.35 (67.7)
LLR	0.2 (0.0)	0.0 (0.0)

Table 12. CPU time for one simulation run in one point in simulations with variable traffic ratio for $L^s = 0$ case.

	CPU time average [s]	CPU time $s^2(S^2)$
MDP	252.2	211.5 (5278.7)
MDP_P	237.3	238.2 (8324.5)
MDP_W	256.1	275.5 (9254.3)
LLR	14.6	10.8 (6.4)

Table 12 shows the average simulation time and pooled variance 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 Figure 10. 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.

The performance measures of interest in the simulations are the reward loss, average call set-up delay and objective reward loss:

$$L = 1 - \frac{\bar{R}}{R} \tag{39}$$

$$\bar{D} = \sum_s \bar{D}^s \frac{\lambda_w^s}{\lambda_w} \tag{40}$$

$$L_D = 1 - \frac{\bar{R}_D}{R} \tag{41}$$

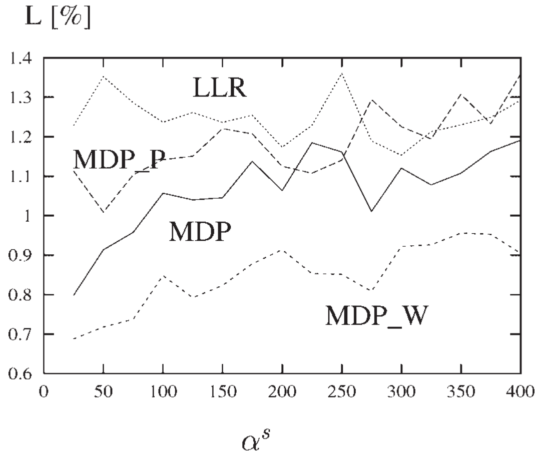

Figure 4. Reward loss of different routing methods versus delay penalty weight.

Figures 4–6 show the loss-delay network routing performance (reward loss, average call set-up delay, objective reward loss) as a function of the delay penalty weight α^s. We have assumed a maximal queue size $L^s = 3$ and a traffic ratio $b_n\lambda_n\mu_n^{-1}/b_w\lambda_w\mu_w^{-1}$ equal to 1 for each OD pair.

Figures 7–9 show the loss-delay network routing performance (reward loss, average call set-up delay, objective reward loss) as a function of the maximal queue size L^s. We have assumed a traffic ratio of 1 for each OD pair and delay penalty weight α^s of 100.

Figure 10 shows the loss network routing performance (reward loss) as a function of the traffic ratio. Different mixes are obtained by varying the per-category call arrival

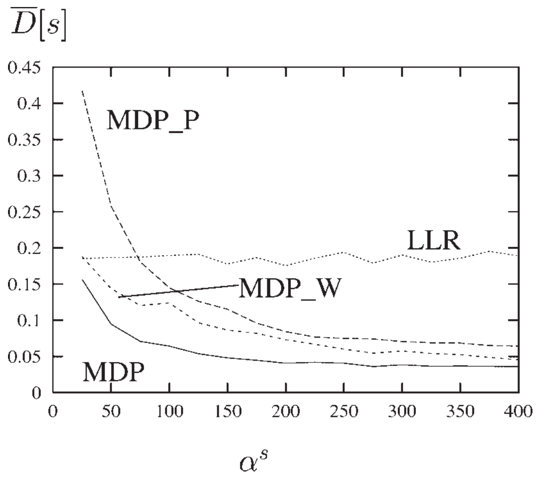

Figure 5. Average call set-up delay of different routing methods versus delay penalty weight.

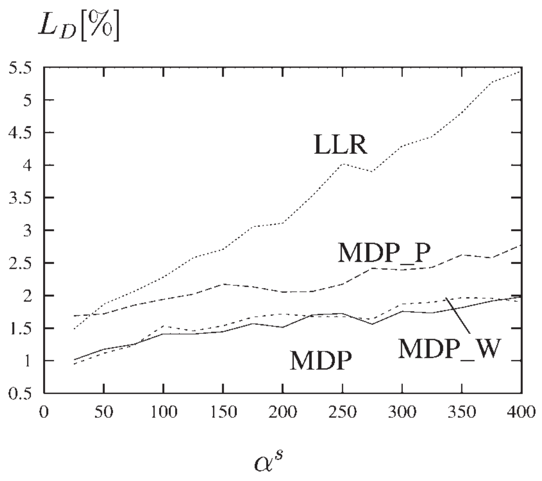

Figure 6. Objective reward loss of different routing methods versus delay penalty weight.

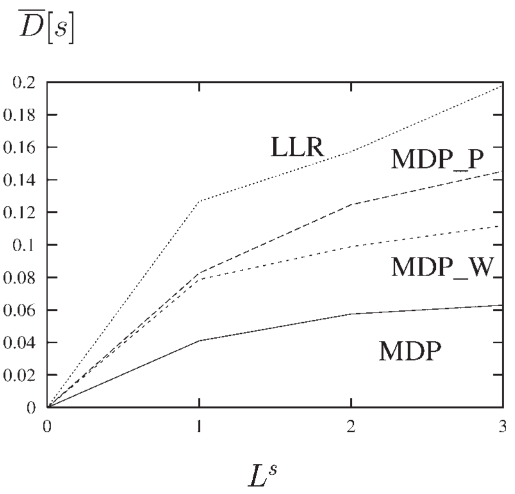

Figure 8. Average call set-up delay of different routing methods versus maximal queue size.

rate to the OD pairs between the simulations, while keeping the amount of traffic per OD pair constant. All OD pairs were offered the same per-category call arrival rates within a simulation.

Figures 11–13 show the loss-delay network routing performance (reward loss, average call set-up delay, objective reward loss) as a function of the traffic ratio. We have assumed a traffic delay penalty weight α^s of 100.

Figures 14–16 show the loss-delay network call blocking probability as a function of the normalised WB reward parameter. The normalised reward parameter, r_j', for class j fulfils $r_j' = r_j\mu_j/b_j$. We have assumed a traffic ratio of 1.0 for each OD pair and a delay penalty weight α^s of 100.

5.3. *Results analysis*

From the graphs in Figures 4–6, which shows the loss-delay network routing performance versus delay penalty weight, the following conclusions are drawn:

- The reward loss for the MDP-based methods tend to increase with the delay penalty weight α^s.
- The reward loss for the LLR method is independent of the penalty weight α^s.
- The average call set-up delay for the MDP-based methods decreases with the delay penalty weight α^s.
- The average call set-up delay for the LLR methods is independent of the delay penalty weight α^s.

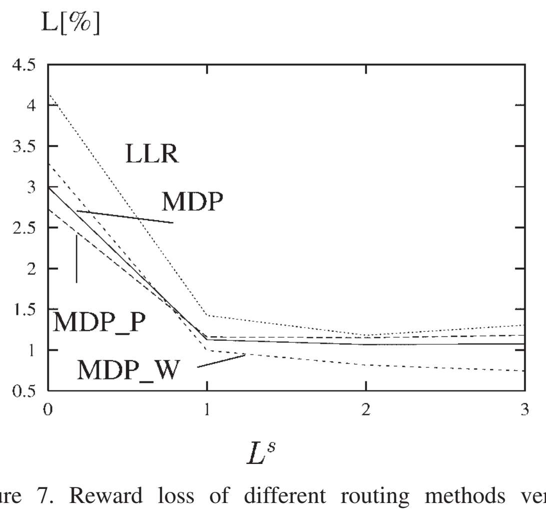

Figure 7. Reward loss of different routing methods versus maximal queue size.

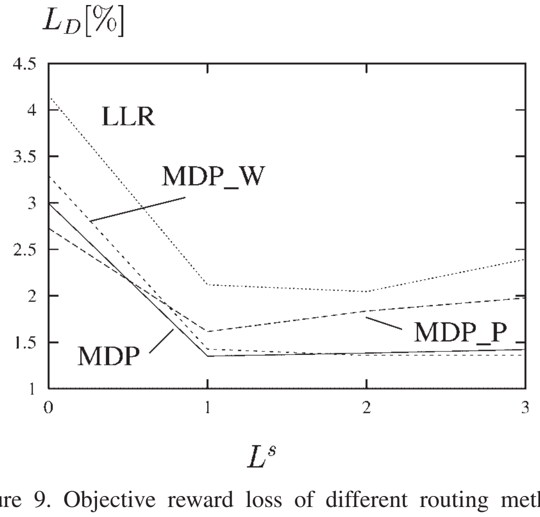

Figure 9. Objective reward loss of different routing methods versus maximal queue size.

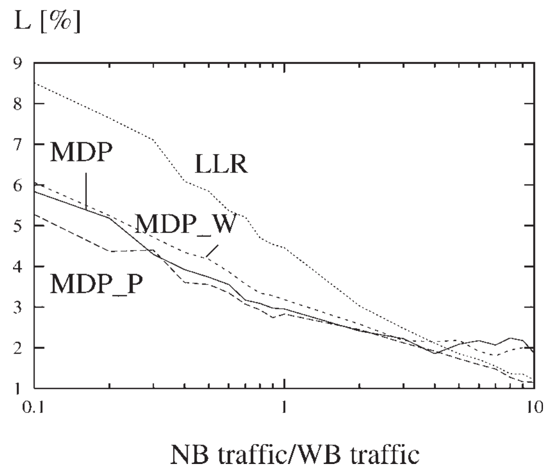

Figure 10. Reward loss of different routing methods versus traffic ratio for $L^s = 0$.

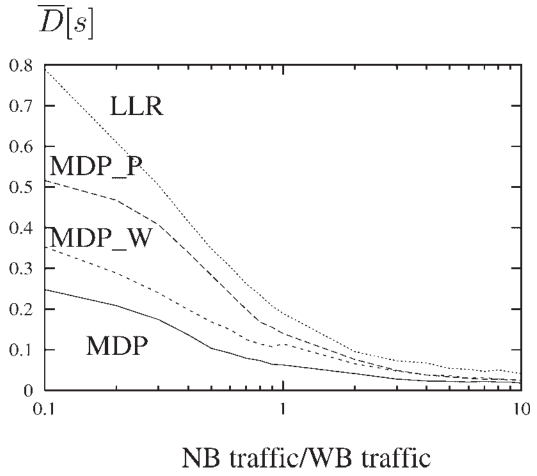

Figure 12. Average call set-up delay of different routing methods versus traffic ratio for $L^s = 3$.

- The objective reward loss increases with the delay penalty weight α^s for all the methods.

From the graphs in Figures 7–9, which show the loss-delay network routing performance versus maximal queue size, the following conclusions are drawn:

- Call queuing reduces the reward loss.
- The average call set-up delay increases when the maximal queue size L^s increases.
- The best overall behaviour in terms of the objective reward loss is obtained for the MDP-based routing algorithms.

From the graphs in Figure 10, which shows the loss network routing performance versus the traffic ratio when $L^s = 0$, the following conclusions are drawn:

- The reward loss decreases when the ratio between NB and WB traffic increases.
- The lowest reward loss is obtained for the MDP-based routing algorithms.

From the graphs in Figures 11–13, which shows the loss-delay network routing performance versus the traffic ratio when $L^s = 3$, the following conclusions are drawn:

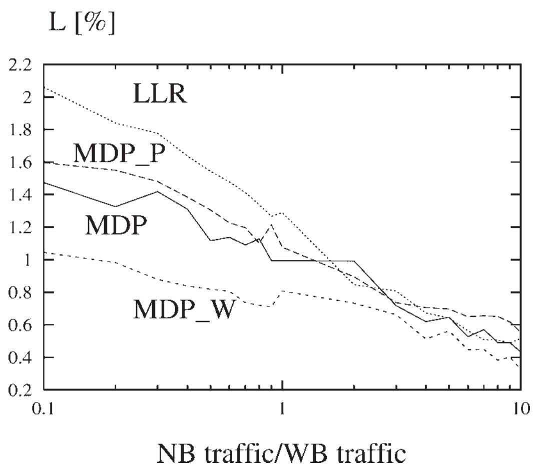

Figure 11. Reward loss of different routing methods versus traffic ratio for $L^s = 3$.

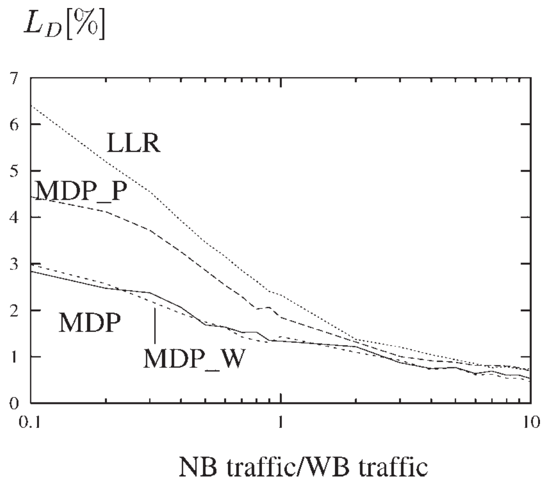

Figure 13. Objective reward loss of different routing methods versus traffic ratio for $L^s = 3$.

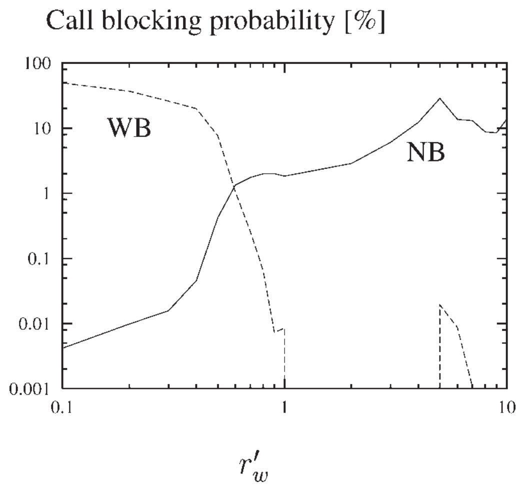

Figure 14. Per-category call blocking probability for MDP method versus normalised reward parameter.

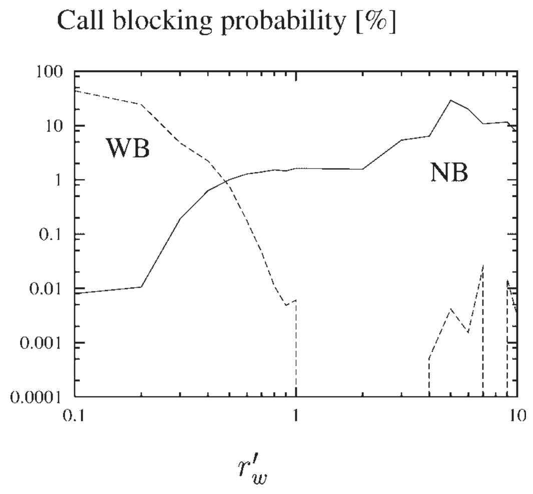

Figure 16. Per-category call blocking probability for MDP_W method versus normalised reward parameter.

- The lowest reward loss is obtained for the MDP-based routing algorithms.
- The reward loss decreases when the ratio between NB and WB traffic increases.
- The average call set-up delay is lowest for the MDP-based routing algorithms.
- The average call set-up delay decreases when the ratio between NB and WB traffic increases.
- The best overall behaviour in terms of the objective reward loss is obtained for the MDP-based routing algorithms.

From the graphs in Figures 14–17, which show the loss-delay network per-category call blocking probability versus normalised WB reward parameter, the following conclusions are drawn:

- The ratio between the NB and WB call blocking probability for the MDP-based methods can be controlled by varying the normalised WB reward parameter.
- The variability in the WB blocking probability for the MDP and MDP_W methods is due to the larger confidence interval for such small values.
- The per-category blocking probabilities of the LLR method are not sensitive to changes in the normalised WB reward parameter

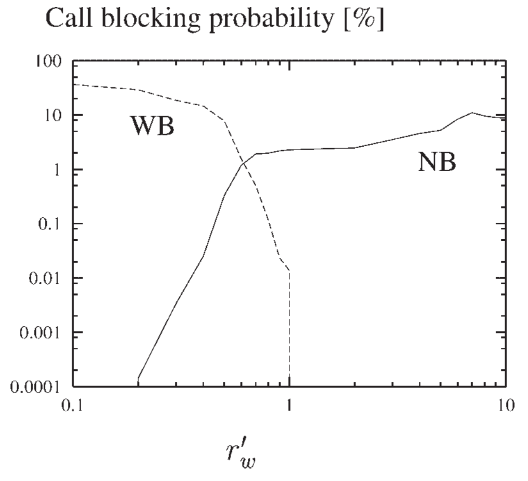

Figure 15. Per-category call blocking probability for MDP_P method versus normalised reward parameter.

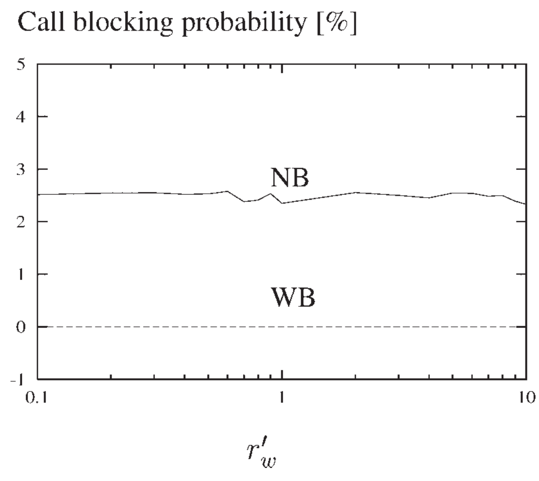

Figure 17. Per-category call blocking probability for LLR method versus normalised reward parameter.

6. CONCLUSION

In this paper, we formulated the CAC and routing problem as a reward maximisation problem with penalty for WB call set-up delay. In this formulation each call class is characterised by its reward parameter defining the expected reward for carrying a call from this class. Such a formulation allows to apply MDP theory to solve the problem.

The computational burden of the exact MDP framework for CAC and routing is prohibitive even for moderate-size networks. Fortunately, it can be reduced to manageable levels by a set of modelling simplifications. First, the network is decomposed into a set of links assumed to have independent traffic and reward processes respectively. Second, the high-dimensional link Markov process and link reward process are aggregated into low-dimensional link Markov process and link reward processes respectively. Third, as will be studied in part II of this paper, the low-dimensional link Markov process and link reward process can be decomposed into per-category link Markov processes and link reward processes.

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 LLR method. The numerical results also showed that the MDP-based methods offer an additional advantage over LLR routing, namely the ability to control the distribution of blocking probabilities among the call classes.

REFERENCES

1. De Serres Y, Mason LG. A multi-server queue, with narrow- and wide-band customers and WB restricted access. *IEEE Transactions on Communications* 1988; **36**(6):675–684.
2. Guérin R. Queuing-blocking system with two arrival streams and guard channels. *IEEE Transactions on Communications* 1988; **36**(2):153–163.
3. Howard RA. *Dynamic Programming and Markov Process*. The M.I.T. Press: Cambridge, Massachusetts, 1960.
4. Tijms H. *Stochastic Modeling and Analysis—A Computational Approach*. John Wiley & Sons: Chichester, 1986.
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, Mignault J, Rosenberg C. Blocking evaluation for networks with reward maximization routing. In *Proceedings of INFOCOM'93*, San Fransisco, USA, 1993.
7. Dziong Z, Mason LG. An analysis of near optimal call admission and routing model for multi-service loss networks. In *Proceedings of IEEE INFOCOM'92*. IEEE Computer Society Press, 1992; pp. 141–152.
8. Dziong Z. *ATM Network Resource Management*. McGraw-Hill: New York, 1997.
9. Liao K-Q, Mason LG. An approximate performance model for a multislot integrated services system. *IEEE Transactions on Communication* 1989; **37**(3):211–221.
10. Tcha D, Jin C, Lutz. Link-by-link bandwidth allocation in an integrated voice/data network using the fuzzy set approach. *Computer Networks and ISDN Systems* 1988/89; **16**:217–227.
11. Zimmermann HJ. Applications of fuzzy set theory to mathematical programming. *Information Science* 1985; **36**:29–58.
12. Dziong Z, Mason LG. Call admission and routing in multi-service loss networks. *IEEE Transactions on Communications* 1994; **42**(2):2011–2022.
13. Ash G. *Dynamic Routing in Telecommunications Networks*. McGraw-Hill: New York, 1998.
14. 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.

AUTHORS' BIOGRAPHIES

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

Zbigniew Dziong received his M.Sc. degree and Ph.D., both in Electrical Engineering from the Warsaw University of Technology, Poland, where he also worked as an assistant professor. During this period, he was on sabbatical leaves at the Centre National d'Etudes des Telecommunications, Paris, France and at the Department of Communication Systems, Lund Institute of Technology, Sweden. From 1987 to 1997, he was with INRS-Telecommunications, Montréal, Canada, as a professor. From 1997 to 2003, he was with Performance Analysis Department at Bell Labs, Lucent Technologies, Holmdel, New Jersey, where he co-authored 14 patents and patent applications. Then he joined École de technologie supérieure, Montréal, Canada, as a professor. His primary research interests cover architecture, performance, management and control issues in optical, data and wireless networks. He was co-recipient of the 1993 STENTOR Award for collaborative research in telecommunications for his contributions in the area of state-dependent routing. He is an author of the book '*ATM Network Resource Management*'.