EUROPEAN TRANSACTIONS ON TELECOMMUNICATIONS
Euro. Trans. Telecomms. (in press)
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/ett.1088

Communication Networks

CAC and routing for multi-service networks with blocked wide-band calls delayed, Part II: approximative 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 with blocked calls cleared and a wide-band with blocked calls delayed. The optimisation is subject to several quality of service (QoS) constraints, either on the packet or call level. The objective function is formulated as reward maximisation with penalty for delay. A suboptimal solution is achieved by applying Markov decision process (MDP) theory together with a three-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. Third, by applying decomposition of the link Markov process, the link MDP tasks are simplified considerably. 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 approximate 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 least loaded routing (LLR).

1. INTRODUCTION

We consider the problem of optimal call admission control (CAC) and routing in multi-service networks such as ATM and STM networks, as well as 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 presentation simplicity, we assume $G = 2$ which is represented by one narrow-band (NB) category requesting a bandwidth of b_n Mbps, and one wide-band (WB) category requesting b_w Mbps $(b_n < b_w)$. The required 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

---

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

Contract/grant sponsors: Swedish Knowledge Foundation; Natural Sciences and Research Council, Canada (NSERC).

Received 21 March 2005
Revised 8 July 2005
Accepted 12 October 2005

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

Modern CAC and routing mechanisms are state-dependent rather than static, which means that the decision to reject the request for a new call, or to accept it on a particular path depends on the current occupancy of the network. The state of the network is represented by the number of calls from each class in service, or waiting for service, at each network link. A state-dependent CAC and routing policy is based on a mapping, for every call class, from a network state space to a set of possible routing decisions, see Figure 1. First, the CAC mechanism 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 CAC and routing mechanism should select one path among the set of feasible paths to convey the call or to reject the request if call acceptance would diminish the expected revenue. While contributing to the 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 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.

Nordström and Dziong 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 maximisation of a reward function being a linear combination of the reward from accepted NB and WB calls and the average WB call set-up delay treated as cost (penalty). A given OD pair can be offered traffic from several NB and WB categories which each can have an unique value of the reward parameter. Since both NB and WB calls are accounted for in the control objective it becomes possible to control the access fairness (call blocking probability) among both the NB and WB classes and not just among the NB classes

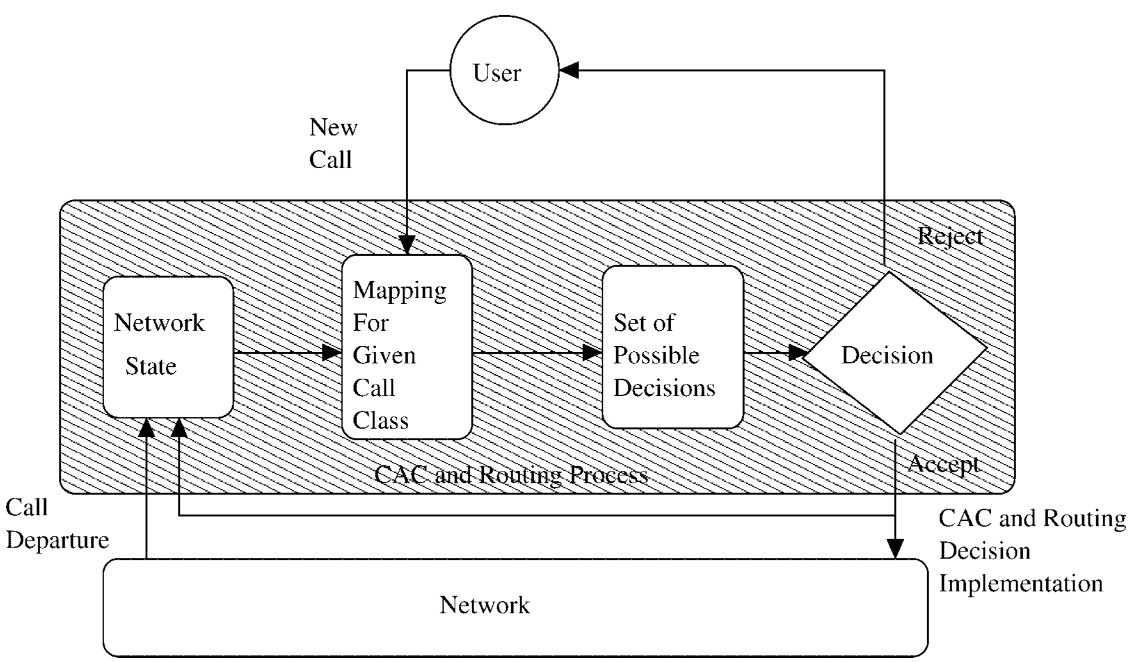


Figure 1. State-dependent CAC and routing.

as was suggested in Reference [6]. The trade-off between NB and WB reward loss and average WB call set-up delay is controlled by the weight of the average delay term.

The computational burden of the exact MDP framework for CAC and routing is prohibitive even for moderate-size networks. Fortunately, it can be reduced to manageable levels by a set of 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 a G dimensional link Markov process and link reward process respectively. Third, as will be studied in this paper, the exact G dimensional link MDP task is transformed into an approximate link MDP task which has feasible computational complexity.

The computational burden of each link MDP task associated with the G dimensional link Markov process increases exponentially with the number of categories G. In order to cope with the problem imposed by large state spaces, several link MDP frameworks with reduced computational cost have been proposed, notably methods based on state aggregation [7], decomposition of the link Markov process [6] and polynomial cost approximation [8, 9].

Krishnan and Hübner proposed a state aggregation link MDP framework based on a scalar link state representing the link occupancy [7]. Transition probabilities between link states were derived from link occupancy probabilities obtained by a recursive procedure due to Kaufman [10] and Roberts [11]. The MDP task was solved by one-step policy iteration.

Nordström and Carlström proposed a modification of the link reward model used in Krishnan's and Hübner's state aggregation method [12]. The new reward model more accurately reflects the bandwidth occupancy by different call categories. The modification can improve the network revenue significantly according to simulation results in Reference [12].

The link Markov process decomposition method is due to Liao and Mason [13] and Dziong, Liao and Mason [6]. They observed that when the holding times of WB calls are significantly longer than for NB calls, the NB process changes state much more often than the WB process. This justifies that the NB and WB process can be analysed separately. The NB process is analysed separately for each state of the WB process, and the WB process is analysed by taking the average 'disturbance' of the NB process into account.

State aggregation cannot be used between the NB loss category and the WB delay category, since the state space is not coordinate convex. The condition for coordinate convexity is not fulfilled since NB transitions are allowed from a given state (due to a NB call departure) but a transition in the other direction (due to a NB call arrival) is not always allowed. In order to maintain coordinate convexity, the NB arrival should sometimes be able to preempt a WB call from the link to the queue, which is not allowed. However, we can use state aggregation within the NB loss category and WB delay category. For example, we can construct an NB super-category from a set of NB subcategories with different NB bandwidth requirements. Link Markov process decomposition can then be applied between the NB and WB super-categories.

Marbach, Mihatch and Tsitsiklis applied reinforcement learning to estimate the optimal second-degree polynomial link-cost approximation [8]. Although the complexity of each simulation step is fixed and low, the required number of simulation steps is large (in the order of 10^7).

Rummukainen and Virtamo proposed an analytical link model for computing the cost relative values as a linear combination of a modest number of basis vectors [9]. Single- and double-coordinate monomial vectors, piecewise constant vectors and piecewise single-coordinate monomial vectors were considered as basis vectors. The MDP task was solved by one-step policy iteration.

In this paper, we propose a new decomposed link MDP model for the CAC and routing objective proposed in part I of this paper [5]. The link MDP model for the WB process is the same as in Reference [6]. The link MDP model for the set of NB processes is new. The set NB processes are classified into NB processes without a trapping state and NB processes with a trapping state. When the NB process enters a trapping state it stays there until the WB process enters a new state. The presence of a trapping state makes the NB process non-irreducible. An NB process without a trapping state can be analysed in the standard way. However, an NB process with a trapping state must use a new solution model to obtain the relative values.

The contribution of this paper is twofold. First, we describe how the concept of link Markov process decomposition can be used to obtain a computationally feasible solution to the considered CAC and routing problem. Second, we present an extensive numerical evaluation, based on simulation, of the MDP-based methods for CAC and routing. For comparison, the performance of the least loaded routing (LLR) algorithm is also evaluated.

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

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 decomposed link MDP model. Section 4 outlines the MDP computation procedure. Section 5 presents formulas for the computational complexity for the approximate link MDP models based on state aggregation and link Markov process decomposition. Section 6 gives a summary of the numerical/simulation-based evaluation of the performance of the MDP-based methods as well as some conventional routing methods. Finally, Section 7 concludes the paper. Sections 2, 3 and 4 have some text common with part I of this paper.

2. CAC AND ROUTING 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, and
- Reward parameter $r_j \in (0, \infty)$.

The parameter r_j is a CAC and routing control parameter that can be used to achieve several different objectives of the network operator. In particular it can be used to 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, $i \in I = \{1, 2\} = \{n, w\}$, on link s, is characterised by:

- Bandwidth requirement $b_i \in \{b_n, b_w\}$ [Mbps],
- 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.

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

Network CAC and routing policy	π
Link CAC policy	π_n^s, π_w^s
Number of OD pairs	P
Number of call classes	K
Number of call categories	G
Class index	j
Category index	i
Link index	s
Path index	k
Set of class indices	J
Set of category indices	I
Set of link indices	S
Set of link indices for path k	S_k
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 link reward parameter	$\bar{r}_i^s(\pi)$
Average mean holding time	$1/\bar{\mu}_i^s$
Offered network reward	R
Carried network objective reward	$\bar{R}_D$
Carried network reward	$\bar{R}$
Average call set-up delay	$\bar{D}$
Delay penalty weight	α
Network state	$\mathbf{z}$
NB link state	$\boldsymbol{x}_n$, $\mathbf{y}_n$
WB link state	$\boldsymbol{x}_w$, $\mathbf{y}_w$
State transformation function	$f_i(\boldsymbol{x})$

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. The following basic scheme of queuing system management, originally proposed in Reference [6], is assumed throughout the paper. When the path chosen by the CAC and routing algorithm has sufficient available capacity for the new WB call, the call is set up between the considered 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 has insufficient capacity on both the link and in the queue. At links with sufficient capacity, bandwidth is reserved for the new WB call while waiting for all links to be ready to accept the call. A link queue is served when a sufficient number of 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

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

NB state space	$X_n(x'_w)$
WB state space	X_w
Set of possible $\boldsymbol{x}'_n$ states	$Y_n(x'_w, x_w)$
Set of possible $\boldsymbol{x}'_w$ states	$Y_w(x'_w)$
Lower limit for NB occupancy	$L(x'_w, x_w)$
Number of class j calls	z_j
Queue state	z_l^s, x_l
Link capacity	C^s
Queue capacity	L^s
NB action space	A_n
WB action space	A_w
Action	a_n, a_w
NB state transition probability	$p_{\boldsymbol{x}_n \boldsymbol{y}_n \mid x'_w}(a_n)$
WB state transition probability	$p_{\boldsymbol{x}_w \boldsymbol{y}_w}(a_w)$
Link reward parameter	$r_j^s(\pi)$
Rate of offered calls	$\lambda_j^k(\pi)$
Rate of accepted calls	$\bar{\lambda}_j^s$
NB link call arrival rate	$\lambda_n^s(\boldsymbol{x}_n, x'_w, \pi)$
WB link call arrival rate	$\lambda_w^s(\boldsymbol{x}_w, \pi)$
Blocking probability	$B_j^c(\pi)$
NB filtering probability	$\phi_{njk}^s(\boldsymbol{x}_n, x'_w, \pi)$
WB filtering probability	$\phi_{wjk}^s(\boldsymbol{x}_w, \pi)$
Category probability	p_{ij}^s
NB average sojourn time	$\tau(\boldsymbol{x}_n, x'_w, a_n)$
WB average sojourn time	$\tau(\boldsymbol{x}_w, a_w)$
NB expected reward	$R_{Dn}^s(\boldsymbol{x}_n, x'_w, a_n)$
WB expected reward	$R_{Dw}^s(\boldsymbol{x}_w, a_w)$
Exceptional service time	$\rho^s(\boldsymbol{x}_w)^{-1}$
Queue access probability	$B_w(x_w)$
NB reward rate	$q_n^s(\boldsymbol{x}_n, x'_w)$
WB reward rate	$q_w^s(\boldsymbol{x}_w)$

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

Path net-gain	$g_j^k(\mathbf{y}, \pi)$
Link net-gain	$g_i^s(\boldsymbol{x}, \pi)$
NB link net-gain	$g_n^s(\boldsymbol{x}_n, x'_w, \pi)$
WB link net-gain	$g_w^s(\boldsymbol{x}_w, \boldsymbol{x}_w^+, \pi)$
NB increment vector	δ_n
WB increment vector	δ_w
Link shadow price	$p_i^s(\boldsymbol{x}, \pi)$
Relative reward	$m^s(y_n, x'_w, a_n)$
NB relative value	$v_n^s(\boldsymbol{x}_n, x_w, \pi)$
WB relative value	$v_w^s(\boldsymbol{x}_w, \pi)$
NB reference state	$\boldsymbol{x}_{nr}$
WB reference state	$\boldsymbol{x}_{wr}$
Trunk reservation parameter	θ_j^n
Total offered traffic load	ρ
#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

on every link along the path for a given WB call, the call is set up between the considered OD node pair.

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

2.3. *QoS and GoS constraints*

CAC and routing faces QoS constraints and, possibly, GoS constraints. First, the CAC_{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 CAC_{GoS} function accepts/rejects this choice based on revenue considerations and end-to-end GoS constraints of the requested call class.

We adopt the definition of QoS and GoS recommended by ITU [14]. 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 offer 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 [15].

2.4. *Objective function*

In our loss-delay type of system, 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 call 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

$$\bar{R} = \sum_j r_j \bar{\lambda}_j \qquad (1)$$

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 [16, 17]. The reward maximisation has several advantages from the management point of view since by controlling the reward parameters one can almost independently control the grade of service of individual streams (cf. [16, 17]). Another advantage is that by using the control model presented in References [16, 17], 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:

$$\bar{R} = \bar{R}_n + \bar{R}_w = \sum_s [\bar{R}^s_n + \bar{R}^s_w] \tag{3}$$

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

$$\bar{D} = \sum_s \bar{D}^s \frac{\lambda^s_w}{\lambda_w} \tag{4}$$

where $\bar{D}^s$ denotes the average delay of calls in the s-th link queue and λ^s_w, λ_w denote 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.

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:

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

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 (5) we arrive at

$$\bar{R}_D = \sum_s \left[\bar{R}^s - \alpha\bar{D}^s \frac{\lambda^s_w}{\lambda_w}\right] \tag{6}$$

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

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

$$\bar{R}_D = \sum_s \left[\bar{R}^s - \alpha^s\bar{D}^s \frac{\lambda^s_w}{\lambda_w}\right] \tag{7}$$

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

3. MDP MODELING

3.1. Network decomposition

The behaviour of the network under consideration can be described by a MDP with the objective to maximise the reward function defined by Equation (7). The corresponding reward rate, $q(\mathbf{z})$, is given by:

$$q(\mathbf{z}) = \sum_{j\in J_n} r_j z_j \mu_n + \sum_{j\in J_w} r_j z_j \mu_w - \sum_{s\in S} \alpha^s \frac{z^s_l}{\lambda_w} \tag{8}$$

where $\mathbf{z}$ denotes the network state and z_j, z^s_l 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

$$A = \{\mathbf{a} = \{a_j\} : a_j \in \{0\} \cup W_j, j \in J\} \tag{9}$$

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-size networks. We therefore decompose the network into a set of links assumed to have independent traffic and reward processes respectively [18].

The network Markov process is decomposed into a set of independent link Markov processes, driven by state-dependent Poisson call arrival processes with rate $\lambda^s_j(\boldsymbol{x}, \pi)$, 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 $r^s_j(\pi)$ fulfil the obvious condition that

$$r_j = \sum_{s\in S_k} r^s_j(\pi) \tag{10}$$

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 [18]. 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 [18]:

$$\bar{r}_i^s(\pi) = \frac{\sum_{j \in J_i} r_j^s(\pi)\bar{\lambda}_j^s(\pi)}{\sum_{j \in J_i} \bar{\lambda}_j^s(\pi)} \tag{11}$$

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. Decomposed link MDP model

It is expected that the considered WB services will be characterised by service times significantly longer than that of NB calls. This feature gives rise for using the so called near-complete decomposability feature. In Reference [13] it was shown that based on this premise one can approximate the link Markov process by two separate NB and WB processes. The presented results indicate that the approach is very efficient and accurate even for the case of equal mean holding times for both NB and WB calls. The link Markov process decomposition for mixed loss-delay call set up was first presented in Reference [6] which considered a different CAC and routing objective than we do. In this subsection, we formulate the decomposed link MDP model for our new CAC and routing objective.

3.2.1. Link model for narrow-band process. 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 $\boldsymbol{x}_n = (x_n, x_w)$, where x_n and x_w denotes the number of NB and WB calls on the link respectively. A Markov NB decision problem is associated with each value x'_w of the number of WB calls in the system.

The state space for the NB process, $\{\boldsymbol{x}_n\}$, in WB state x'_w, is given by:

$$X_n(x'_w) = \{\boldsymbol{x}_n = (x_n, x_w) : x_n \in Y_n(x'_w, x_w) \text{ and } x_w \in Y_w(x'_w)\} \tag{12}$$

where the set $Y_n(x'_w, x_w)$ denotes the possible values of x_n when the number of WB calls in the system is x'_w and the number of WB calls on the link is x_w:

$$Y_n(x'_w, x_w) = \left\{x_n : L(x'_w, x_w) \le x_n b_n \le C^s - x_w b_w\right\} \tag{13}$$

where

$$L(x'_w, x_w) = \begin{cases} 0, & x'_w = x_w \\ C^s - (x_w+1)b_w, & x'_w > x_w, x_w < N_w^s \\ 0, & x'_w > x_w, x_w = N_w^s \end{cases} \tag{14}$$

where N_w^s denotes the maximum number of WB calls on link s. The set $Y_w(x'_w)$ denotes the possible states x_w in WB state x'_w:

$$Y_w(x'_w) = \left\{x_w : \max(0, x'_w - L^s) \le x_w \le \min(N_w^s, x'_w)\right\} \tag{15}$$

where L^s denotes the maximum queue size on link s. The action space is given by:

$$A_n = \{a_n : a_n \in \{0, 1\}\} \tag{16}$$

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

$$A_n(\boldsymbol{x}_n, x'_w) = \left\{a_n \in A_n : a_n = 0 \text{ if } \boldsymbol{x}_n + \delta_n \notin X_n(x'_w)\right\} \tag{17}$$

where δ_n denotes the vector $(1,0)^{\mathrm{T}}$. The state transition probabilities are given by:

$$p_{\boldsymbol{x}_n \mathbf{y}_n | x'_w}(a_n) = \begin{cases} \lambda_n^s(\boldsymbol{x}_n, x'_w, \pi) a_n \tau(\boldsymbol{x}_n, x'_w a_n), & \mathbf{y}_n = \boldsymbol{x}_n + \delta_n \in X_n(x'_w) \\ x_n \bar{\mu}_n^s \tau(\boldsymbol{x}_n, x'_w, a_n), & \mathbf{y}_n = \boldsymbol{x}_n - \delta_n \in X_n(x'_w) \text{ and } \boldsymbol{x}_n - \delta_n + \delta_w \notin X_n(x'_w) \\ x_n \bar{\mu}_n^s \tau(\boldsymbol{x}_n, x'_w, a_n), & \mathbf{y}_n = \boldsymbol{x}_n - \delta_n + \delta_w \in X_n(x'_w) \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

where $\lambda_n^s(\boldsymbol{x}_n, x'_w, \pi)$ denotes the NB category arrival rate to the link in state $(\boldsymbol{x}_n, x'_w)$ under routing policy π, $\tau(\boldsymbol{x}_n, x'_w, a_n)$ denotes the average sojourn time in state $(\boldsymbol{x}_n, x'_w)$ and δ_w denotes the vector $(0, 1)^{\mathrm{T}}$.

The link call arrival rates, $\lambda_n^s(\boldsymbol{x}_n, x'_w, \pi)$ are given by:

$$\lambda_n^s(\boldsymbol{x}_n, x'_w, \pi) = \sum_{j \in J_n} \sum_{k \in W_j^s} \lambda_j^k(\pi) \phi_{njk}^s(\boldsymbol{x}_n, x'_w, \pi) \prod_{c \in S_k \setminus \{s\}} (1 - B_j^c(\pi)) \quad (19)$$

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 $\phi_{njk}^s(\boldsymbol{x}_n, x'_w, \pi)$ denotes a filtering probability defined as:

$$\phi_{njk}^s(\boldsymbol{x}_n, x'_w, \pi) = P\left\{ \sum_{c \in S_k \setminus \{s\}} p_j^c(\boldsymbol{x}_n, x'_w, \pi) < r_j - p_j^s(\boldsymbol{x}_n, x'_w, \pi) | \bar{\boldsymbol{B}}_j \right\} \quad (20)$$

where $\bar{\boldsymbol{B}}_j$ denotes the condition that no link on path k is in the blocking state, $p_j^c(\boldsymbol{x}_n, x'_w, \pi)$ denotes the state-dependent *link shadow price* for class j on link c (note that $p_j^s(\boldsymbol{x}_n, x'_w, \pi)$ is constant in Equation (20)). In other words $\phi_{njk}^s(\boldsymbol{x}_n, x'_w, \pi)$ 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 [15], or approximated with one according to experiments in Reference [18]. The $\lambda_j^k(\pi)$ denote the arrival rate of class j to path $k \in W_j$, and is given by the following load sharing model [18]:

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

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:

$$\bar{\mu}_n^s = \left[\sum_{j \in J_n} p_{nj}^s \mu_j^{-1} \right]^{-1} \quad (22)$$

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

$$p_{nj}^s = \frac{\bar{\lambda}_j^s(\pi)}{\sum_{c \in J_n} \bar{\lambda}_c^s(\pi)} \quad (23)$$

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

The average sojourn time in state $(\boldsymbol{x}_n, x'_w)$ is given by:

$$\tau(\boldsymbol{x}_n, x'_w, a_n) = \left\{ x_n \bar{\mu}_n^s + a_n \lambda_n^s(\boldsymbol{x}_n, x'_w, \pi) \right\}^{-1} \quad (24)$$

The expected reward in state $(\boldsymbol{x}_n, x'_w)$ is given by $R_{Dn}^s(\boldsymbol{x}_n, x'_w, a_n) = q_n^s(\boldsymbol{x}_n, x'_w)\tau(\boldsymbol{x}_n, x'_w, a_n)$, where $q_n^s(\boldsymbol{x}_n, x'_w)$ is obtained from

$$q_n^s(\boldsymbol{x}_n, x'_w) = \bar{r}_n^s(\pi) x_n \bar{\mu}_n^s + \bar{r}_w^s(\pi) x_w \bar{\mu}_w^s - \alpha^s \frac{x_l}{\lambda_w} \quad (25)$$

where $\boldsymbol{x}_n = (x_n, x_w)$ and $x_l = x'_w - x_w$.

3.2.2. Link MDP model for the wide-band process. The state of the WB link process can be described by a vector $\boldsymbol{x}_w = (x'_w, x_w)$, where x'_w denotes the number of WB calls in the system (on the link and in the queue) and x_w denotes the number of WB calls on the link. The state space for the WB process is:

$$X_w = \left\{ \boldsymbol{x}_w = (x'_w, x_w) : x'_w \in Y'_w \text{ and } x_w \in Y_w(x'_w) \right\} \quad (26)$$

where

$$Y'_w = \left\{ x'_w : 0 \le x'_w \le N_w^s + L^s \right\} \quad (27)$$

and $Y_w(x'_w)$ is defined by expression (15) in the previous subsection. The action space is given by:

$$A_w = \{a_w : a_w \in \{0, 1\}\} \quad (28)$$

The permissible action space is a state-dependent subset of A_w:

$$A_w(\boldsymbol{x}_w) = \left\{ a_w \in A_w : a_w = 0 \text{ if } x'_w + 1 \notin Y'_w \right\} \quad (29)$$

The state transition probabilities when the queue is empty $(x_l = 0)$ are given by:

$$p_{\boldsymbol{x}_w \boldsymbol{y}_w}(a_w) = \begin{cases} \lambda_w^s(\boldsymbol{x}_w, \pi) * & y_w = x_w + 1 \in Y_w(x'_w) \\ (1 - B_w(x_w)) a_w \tau(\boldsymbol{x}_w, a_w), & y'_w = x'_w + 1 \in Y'_w \\ \lambda_w^s(\boldsymbol{x}_w, \pi) * & y_w = x_w \in Y_w(x'_w) \\ B_w(x_w) a_w \tau(\boldsymbol{x}_w, a_w), & y'_w = x'_w + 1 \in Y'_w \\ x_w \bar{\mu}_w^s \tau(\boldsymbol{x}_w, a_w), & y_w = x_w - 1 \in Y_w(x'_w) \\ & y'_w = x'_w - 1 \in Y'_w \end{cases} \quad (30)$$

where $\lambda_w^s(\boldsymbol{x}_w, \pi)$ is given by a formula analogous to Equation (19), and $B_w(x_w)$ denotes the probability that an arriving WB call will enter the queue; for details of the computation procedure, see Reference [13]. The state transition probabilities when the queue in non-empty ($0 < x_l \le L^s$) are given by:

$$p_{\boldsymbol{x}_w \mathbf{y}_w}(a_w) = \begin{cases} \lambda_w^s(\boldsymbol{x}_w, \pi) a_w \tau(\boldsymbol{x}_w, a_w), & y_w = x_w \in Y_w(x'_w) \\ & y'_w = x'_w + 1 \in Y'_w \\ x_w \bar{\mu}_w^s \tau(\boldsymbol{x}_w, a_w), & y_w = x_w \in Y_w(x'_w) \\ & y'_w = x'_w - 1 \in Y'_w \\ \rho^s(\boldsymbol{x}_w)\tau(\boldsymbol{x}_w, a_w), & y_w = x_w + 1 \in Y_w(x'_w) \\ & y'_w = x'_w \in Y'_w \end{cases} \tag{31}$$

where $\rho_s(\boldsymbol{x}_w)^{-1}$ denotes the exceptional service time. This time is equal to the passage time of the NB process from the state $\boldsymbol{x}_n = (x_n, x_w)$ at the moment of entering $\boldsymbol{x}_w = (x'_w, x_w)$, to the state $\boldsymbol{x}'_n = (x'_n, x_w)$, where $x'_n = C^s - (x_w + 1)b_w$. The exceptional service time has a complex distribution. To cope with the problem we apply an approximation presented in Reference [13] where the distribution was approximated by an exponential distribution with mean $\rho_s(\boldsymbol{x}_w)^{-1}$. For details of the evaluation, see Reference [13].

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

$$\bar{\mu}_w^s = \left[\sum_{j \in J_w} p_{wj}^s \mu_j^{-1} + T^s \right]^{-1} \tag{32}$$

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

The average sojourn time in state $(\boldsymbol{x}_w)$ is given by:

$$\tau(\boldsymbol{x}_w, a_w) = \{x_w \bar{\mu}_w^s + \rho^s(\boldsymbol{x}_w) + a_w \lambda_w^s(\boldsymbol{x}_w, \pi)\}^{-1} \tag{33}$$

The expected reward in state $\boldsymbol{x}_w$ is given by $R_{Dw}^s(\boldsymbol{x}_w, a_w) = q_w^s(\boldsymbol{x}_w)\tau(\boldsymbol{x}_w, a_w)$, where $q_w^s(\boldsymbol{x}_w)$ is obtained from

$$q_w^s(\boldsymbol{x}_w) = \bar{r}_n^s(\pi)\bar{x}_n(\boldsymbol{x}_w)\bar{\mu}_n^s + \bar{r}_w^s(\pi) x_w \bar{\mu}_w^s - \alpha^s \frac{x_l}{\lambda_w} \tag{34}$$

where $\bar{x}_n(\boldsymbol{x}_w)$ denotes the average number of NB calls present on the link in WB state $\boldsymbol{x}_w$. Note that $\bar{x}_n(\boldsymbol{x}_w)$ can easily be evaluated from flow balance equations for the NB process.

4. MDP COMPUTATIONAL PROCEDURE

This section outlines the MDP computational procedure for determining a near-optimal CAC and routing policy using the decomposed link model. The central idea is to compute *path net-gain* functions, $g_j^k(\mathbf{y}, \pi)$, which estimate the increase in long-term reward due to admission of a class j call on path k in network state $\mathbf{y}$. The CAC and routing rule is simply to choose, given the state of the network and the class of the call request, a path which offer 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.

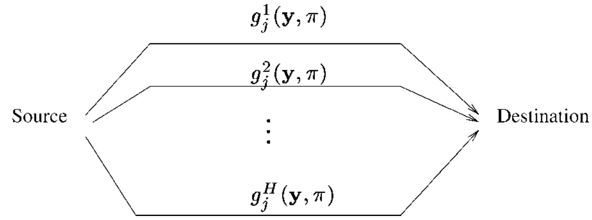


Figure 2. The call is offered to a path which has sufficient QoS and maximal positive path net-gain among the $H = |W_j|$ alternative paths.

4.1. Basic definitions

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

$$g_j^k(\mathbf{y}, \pi) = r_j - \sum_{s \in S_k} p_i^s(\boldsymbol{x}, \pi) \tag{35}$$

where traffic class j is a member of traffic category i, $\mathbf{y} = \{\boldsymbol{x}\}$ denotes the network state in the exact network model, and $p_i^s(\boldsymbol{x}, \pi)$ 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 $\boldsymbol{x} = (x_n, x'_w)$ and is defined as follows:

$$p_i^s(\boldsymbol{x}, \pi) = \bar{r}_i^s(\pi) - g_i^s(\boldsymbol{x}, \pi) \tag{36}$$

where $g_i^s(\boldsymbol{x}, \pi)$ denotes the *link net-gain* for admission of a category-i call in state $\boldsymbol{x}$. The link net-gain expresses the increase in long-term reward due to admission of a category-i call in link state $\boldsymbol{x}$. For the decomposed link model, the link net-gain for the NB category is defined as:

$$\begin{aligned} g_n^s(\mathbf{f}_n(\boldsymbol{x}), x'_w, \pi) &= g_n^s(\boldsymbol{x}_n, x'_w, \pi) \\ &= v_n^s(\boldsymbol{x}_n + \delta_n, x'_w, \pi) - v_n^s(\boldsymbol{x}_n, x'_w, \pi) \end{aligned} \tag{37}$$

where $\boldsymbol{x}_n$ denotes the state of the NB process in WB state x'_w. The NB state is obtained from the exact link state as follows:

$$\boldsymbol{x}_n = \mathbf{f}_n(\boldsymbol{x}) = (x_n, x'_w - x_l) \tag{38}$$

where x_l is given by:

$$x_l = f_l(\boldsymbol{x}) := \inf\{x_l : x_l \geq 0, C^s - x_n b_n \geq (x'_w - x_l) b_w\} \tag{39}$$

The link net-gain for the WB category is defined as:

$$\begin{aligned} &g^s_w(\mathbf{f}_w(\boldsymbol{x}), \mathbf{f}_w(\boldsymbol{x} + \delta_w), \pi) \\ &= g^s_w(\boldsymbol{x}_w, \boldsymbol{x}^+_w, \pi) = v^s_w(\boldsymbol{x}^+_w, \pi) - v^s_w(\boldsymbol{x}_w, \pi) \end{aligned} \tag{40}$$

where the $\boldsymbol{x}_w$ denotes the state of the WB process before admission, $\boldsymbol{x}^+_w$ denotes the state of the WB process after admission. The WB state is obtained from the exact link state as follows:

$$\boldsymbol{x}_w = \mathbf{f}_w(\boldsymbol{x}) = (x'_w, x'_w - x_l) \tag{41}$$

To give more insight into the definition of relative values, let us define the expected link reward, $R^s_D(\boldsymbol{x}_0, \pi, T)$, obtained in a interval $(t_0, t_0 + T)$ of length T, assuming state $\boldsymbol{x}_0$ at time t_0:

$$R^s_D(\boldsymbol{x}_0, \pi, T) = E\left[\int_{t_0}^{t_0+T} q^s(\boldsymbol{x}(t))\mathrm{d}t\right] \tag{42}$$

where $q^s(\boldsymbol{x}(t))$ denotes the expected reward accumulation rate in state $\boldsymbol{x}(t)$. The process $\{\boldsymbol{x}(t)\}$ is driven by a probabilistic law of motion specified by certain state transition probabilities. The relative value can now be written as:

$$v^s(\boldsymbol{x}_0, \pi) = \lim_{T\to\infty} \left[R^s_D(\boldsymbol{x}_0, \pi, T) - R^s_D(\boldsymbol{x}_r, \pi, T)\right] \tag{43}$$

That is, the relative value in state $\boldsymbol{x}_0$ is defined as the difference in future reward earnings when starting in the given state, compared to a reference state, $\boldsymbol{x}_r$. In practice, the relative value function is obtained by solving a set of linear equations (see below).

4.2. Adaptation of the CAC and routing policy

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

1. *Startup*: Initialise the relative values in a way that make all link net-gains with permissible admission positive.
2. *On-line operation phase*: Measure per-path call acceptance rates $\bar{\lambda}^k_j(\pi)$ and per-link blocking probabilities $B^c_j(\pi)$ 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:
   NB link model: determine link call arrival rates $\lambda^s_n(\boldsymbol{x}_n, x'_w, \pi)$ and per-category reward parameters $\bar{r}^s_n(\pi)$.
   WB link model: determine link call arrival rates $\lambda^s_w(\boldsymbol{x}_w, \pi)$ and per-category reward parameters $\bar{r}^s_w(\pi)$.
   (b) Value determination:
   NB link model: find relative values $v^s_n(\boldsymbol{x}_n, x'_w, \pi)$ and average reward rate $\bar{R}^s_{Dn}(\pi)$ for the current policy π.
   WB link model: find relative values $v^s_w(\boldsymbol{x}_w, \pi)$ and average reward rate $\bar{R}^s_{Dw}(\pi)$ for the current policy π.
   (c) Policy improvement: Find the new link CAC policies $\pi^{s\prime}_n$ and $\pi^{s\prime}_w$ based on the new relative values and the new average reward rates.
4. *Convergence test*: For each category, repeat from 2 until the 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 [4].

4.2.1. Value determination for the NB link model. For the NB link model, different solution techniques are used depending on whether the NB process contains a trapping state or not. When $x'_w < N^s_w$, the NB process does not contain a trapping state. In this case, the following sparse system of linear equations are should be solved to obtain the relative values and the average reward rate:

$$\begin{cases} v^s_n(\boldsymbol{x}_n, x'_w, \pi) = R^s_{Dn}(\boldsymbol{x}_n, x'_w, a_n) - \bar{R}^s_{Dn}(\pi)\tau(\boldsymbol{x}_n, x'_w, a_n) + \sum_{\mathbf{y}_n \in X_n(x'_w)} p_{\boldsymbol{x}_n \mathbf{y}_n | x'_w}(a_n) v^s_n(\mathbf{y}_n, x'_w, \pi) \\ v^s_n(\boldsymbol{x}_{nr}, x'_w, \pi) = 0; \qquad \boldsymbol{x}_{nr} \in X_n(x'_w), \boldsymbol{x}_n \in X_n(x'_w) \backslash \{\boldsymbol{x}_{nr}\} \end{cases} \tag{44}$$

where the following quantities need to be specified:

- $X_n(x'_w)$: the NB state space given WB state x'_w,
- $a_n = \pi^s_n(\boldsymbol{x}_n, x'_w)$: the control action in NB state $\boldsymbol{x}_n$ given WB state x'_w,

- $\tau(\boldsymbol{x}_n, x'_w, a_n)$: the expected sojourn time in NB state $\boldsymbol{x}_n$ given WB state x'_w,
- $R^s_{Dn}(\boldsymbol{x}_n, x'_w, a_n)$: the expected link reward when leaving NB state $\boldsymbol{x}_n$ given WB state x'_w,
- $p_{\boldsymbol{x}_n\mathbf{y}_n|x'_w}(a_n)$: the transition probability from NB state $\boldsymbol{x}_n$ to NB state state $\mathbf{y}_n$, given that action a_n is taken in NB state $\boldsymbol{x}_n$, given WB state x'_w,
- $\boldsymbol{x}_{nr}$: the reference NB state,

in order to compute the unknowns:

- $v^s_n(\boldsymbol{x}_n, x'_w, \pi)$: the relative value in NB state $\boldsymbol{x}_n$ under routing policy π given WB state x'_w,
- $\bar{R}^s_{Dn}(\pi)$: the average rate of NB link reward under policy π.

A special solution technique is needed when the NB process contains a trapping state, which occurs when $N^s_w \leq x'_w \leq N^s_w + L^s$. When the NB process enters a trapping state it stays there until the WB process enters a new state. The trapping state of the NB process is $\boldsymbol{x}_{nr} = (0, N^s_w)$. Moreover, the average reward rate is given by $\bar{R}^s_{Dn}(\pi) = q(\boldsymbol{x}_{nr}, x'_w)$.

Let us define the *relative reward rate* in state $\boldsymbol{x}_n$ as the difference

$$\rho^s_{Dn}(\boldsymbol{x}_n, x'_w, \pi) = q^s_n(\boldsymbol{x}_n, x'_w) - \bar{R}^s_{Dn}(\pi) \tag{45}$$

where $q^s_n(\boldsymbol{x}_n, x'_w)$ denotes the immediate NB reward rate in state $\boldsymbol{x}_n$ and $\bar{R}^s_{Dn}(\pi)$ denotes the average NB reward rate.

The expected accumulated NB relative reward in the interval $(t_0, t_0 + T)$ is given by

$$\Delta^s_{Dn}(\boldsymbol{x}_n, x'_w, \pi, T) = E\left[\int_{t_0}^{t_0+T} \rho^s_{Dn}(\mathbf{y}_n(t), x'_w, \pi)\mathrm{d}t\right] \tag{46}$$

where the initial state is given by $\mathbf{y}_n(t_0) = \boldsymbol{x}_n$. The relative value $v^s_n(\boldsymbol{x}_n, x'_w, \pi)$ can be interpreted as the expected accumulated relative reward over an infinite interval when starting from state $\boldsymbol{x}_n$:

$$v^s_n(\boldsymbol{x}_n, x'_w, \pi) = \lim_{T\to\infty} \Delta^s_{Dn}(\boldsymbol{x}_n, x'_w, \pi, T) \tag{47}$$

The definitions (45),(46) and (47) apply to Markov processes with or without a trapping state. Now consider specifically the NB process with a trapping state. First observe that the state-dependent relative reward rate becomes zero when the trapping state is reached. Let $m^s(y_n, x'_w, a_n)$ represent the accumulated relative reward obtained by accumulating relative rewards from the starting state y_n until reaching the adjacent lower state $y_n - 1$. We define the relative value in state $\boldsymbol{x}_n$ for a NB process with a trapping state as the sum of accumulated relative rewards from the starting state $\boldsymbol{x}_n$ until reaching the trapping state $\boldsymbol{x}_{nr}$:

$$\begin{cases} v^s_n(\boldsymbol{x}_n, x'_w, \pi) = \sum_{y_n=1}^{x_n} m^s(y_n, x'_w, \pi^s_n(y_n, f_w(y_n), x'_w)); \quad \boldsymbol{x}_n \in X_n(x'_w)\backslash\{0, N^s_w\}, \\ v^s_n((0, N^s_w), x'_w, \pi) = 0 \end{cases} \tag{48}$$

where $\mathbf{y}_n = (y_n, y_w)$ and $y_w = f_w(y_n) = \sup(y_w : C^s - y_n b_n \geq y_w b_w)$.

Lets assume the accumulated relative reward is delivered to the network when the NB process eventually makes a transition from state y_n to the adjacent lower state $y_n - 1$. The rate of delivered accumulated relative reward is given by $m^s(y_n, x'_w, a_n)y_n\bar{\mu}^s_n$. Note that this accumulated relative reward is a sum of relative reward obtained in state y_n and the accumulated relative reward obtained from the upper adjacent state $y_n + 1$. Hence, we obtain the following recursive formula for the accumulated relative reward values $m^s(y_n, x'_w, a_n)$:

$$\begin{cases} m^s(y_n, x'_w, a_n)y_n\bar{\mu}^s_n = \rho^s_{Dn}(y_n, f_w(y_n), x'_w, \pi) + \\ \lambda^s_n(y_n, f_w(y_n), x'_w, \pi)a_n m^s(y_n + 1, x'_w, \pi^s_n(y_n + 1, f_w(y_n + 1), x'_w)), \quad y_n = 1, \ldots, y^{\max}_n - 1 \\ m^s(y^{\max}_n, x'_w, a_n)y^{\max}_n\bar{\mu}^s_n = \rho^s_{Dn}(y^{\max}_n, f_w(y^{\max}_n), x'_w, \pi) \end{cases} \tag{49}$$

where $y^{\max}_n = C^s - x_w b_w$ denotes the maximum number of NB calls that can be accepted on the link, given that x_w WB calls currently are active on the link. Liao and Mason proposed in Reference [13] a similar recursive formula for the first passage time between Markov states.

4.2.2. Value determination for the WB link model. For the WB category, the following sparse system of linear equations should be solved in order to obtain the relative values and the average reward rate:

$$\begin{cases} v_w^s(\boldsymbol{x}_w, \pi) = R_{Dw}^s(\boldsymbol{x}_w, a_w) - \bar{R}_{Dw}^s(\pi)\tau(\boldsymbol{x}_w, a_w) + \sum_{\mathbf{y}_w \in X_w} p_{\boldsymbol{x}_w \mathbf{y}_w}(a_w) v_w^s(\mathbf{y}_w, \pi) \\ v_w^s(\boldsymbol{x}_{wr}, \pi) = 0; \quad \boldsymbol{x}_{wr} \in X_w, \boldsymbol{x}_w \in X_w \backslash \{\boldsymbol{x}_{wr}\} \end{cases} \tag{50}$$

where the following quantities need to be specified:

- X_w: the WB state space,
- $a_w = \pi_w^s(\boldsymbol{x}_w)$: the control action in WB state $\boldsymbol{x}_w$,
- $\tau(\boldsymbol{x}_w, a_w)$: the expected sojourn time in WB state $\boldsymbol{x}_w$,
- $R_{Dw}^s(\boldsymbol{x}_w, a_w)$: the expected link reward when leaving WB state $\boldsymbol{x}_w$,
- $p_{\boldsymbol{x}_w \mathbf{y}_w}(a_w)$: the transition probability from WB state $\boldsymbol{x}_w$ to WB state $\mathbf{y}_w$, given that action a_w is taken in WB state $\boldsymbol{x}_w$,
- $\boldsymbol{x}_{wr}$: the reference WB state,

in order to compute the unknowns:

- $v_w^s(\boldsymbol{x}_w, \pi)$: the relative value in WB state $\boldsymbol{x}_w$ under routing policy π,
- $\bar{R}_{Dw}^s(\pi)$: the average rate of WB link reward under policy π.

4.2.3. Policy improvement for NB link model. For the NB link model, in case the NB process does not contain any trapping state ($x'_w < N_w^s$) the policy improvement step becomes:

$$a_n = \max_{u_n \in A_n(\boldsymbol{x}_n, x'_w)} \Bigg\{ R_{Dn}^s(\boldsymbol{x}_n, x'_w, u_n) - \bar{R}_{Dn}^s(\pi)\tau(\boldsymbol{x}_n, x'_w, u_n) + \sum_{\mathbf{y}_n \in X_n(x'_w)} p_{\boldsymbol{x}_n \mathbf{y}_n}(u_n) v_n^s(\mathbf{y}_n, x'_w, \pi) \Bigg\} \tag{51}$$

In case the NB process does contain a trapping state ($N_w^s \leq x'_w \leq N_w^s + L^s$) the policy improvement step becomes:

$$a_n = \max_{u_n \in A_n(\boldsymbol{x}_n, x'_w)} \left\{ m^s(x_n, x'_w, u_n) \right\} \tag{52}$$

4.2.4. Policy improvement for WB link model. For the WB link model the policy improvement step becomes:

$$a_w = \max_{u_w \in A_w(\boldsymbol{x}_w)} \Bigg\{ R_{Dw}^s(\boldsymbol{x}_w, u_w) - \bar{R}_{Dw}^s(\pi)\tau(\boldsymbol{x}_w, u_w) + \sum_{\mathbf{y}_w \in X_w} p_{\boldsymbol{x}_w \mathbf{y}_w}(u_w) v_w^s(\mathbf{y}_w, \pi) \Bigg\} \tag{53}$$

5. COMPUTATIONAL COMPLEXITY

5.1. Complexity of policy iteration

The computation (time) complexity of policy iteration is shown in Table 4. In the table, we have assumed that traditional Gauss elimination is used to find the relative values, in which case the complexity is cubic in the size of the state space. 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.

5.2. State space cardinality

5.2.1. Exact link model

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

$$S = \sum_{x'_w=0}^{N_w^s + L^s} N_n(x'_w) \tag{54}$$

Table 4. Computational complexity of policy iteration.

	Value determination	Policy improvement
Exact link model	S^3	$S2^G$
NB link model	$\sum_{x'_w=0}^{N_w^s+L^s} S_n(x'_w)^3$	$\sum_{x'_w=0}^{N_w^s+L^s} 2S_n(x'_w)$
WB link model	S_w^3	$2S_w$
State aggregation link model	S_a^3	$S_a 2^G$

where $N_n(x'_w)$ denotes the number of NB states in the state space when x'_w WB calls are present in the system:

$$N_n(x'_w) = C^s - (x'_w - x_l)b_w + 1 \tag{55}$$

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

$$S \sim \left\{ \frac{1}{G!} \prod_{i \in I} S_i \right\} + L^s \{ \lfloor C^s / b_n \rfloor + 1 \} \tag{56}$$

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.

5.2.2. NB link model. The state of the NB link process can be described by a vector $\boldsymbol{x}_n = (x_n, x_w)$, where x_n and x_w denotes the number of NB and WB calls, respectively, on the link. 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 size of the NB state space in WB state x'_w is given by:

$$S_n(x'_w) = \sum_{x_w = \max(0, x'_w - L^s)}^{\min(N^s_w, x'_w)} N_n(x'_w, x_w) \tag{57}$$

where $N_n(x'_w, x_w)$ denotes the number of NB states in the state space when the number of WB calls in the system is x'_w and the number of WB calls on the link is x_w:

$$N_n(x'_w, x_w) = \begin{cases} C^s - x_w b_w + 1, & x'_w = x_w \\ b_w, & x'_w > x_w, x_w < N^s_w \\ 1, & x'_w > x_w, x_w = N^s_w \end{cases} \tag{58}$$

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

$$S_w = \sum_{x'_w = 0}^{N^s_w + L^s} N_w(x'_w) \tag{59}$$

where $N_w(x'_w)$ denotes the number of WB states in the state space when the number of WB calls in the system is x'_w:

$$N_w(x'_w) = \min(N^s_w, x'_w) - \max(0, x'_w - L^s) + 1 \tag{60}$$

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

6. NUMERICAL RESULTS

6.1. 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 MDP-based routing algorithms are compared:

- MDP—MDP routing based on exact link model proposed in Subsection 3.2 in part I of this paper [5],
- MDP_D—MDP routing based on the decomposed link model proposed in Subsection 3.2 in part II of this paper,
- MDP_A—MDP routing based on Krishnan's and Hübner's state aggregation link model [7] with modified link reward parameters [12].

The two standard simplifications of the MDP model, that is, link independence and aggregation of call classes into call categories are applied by all the MDP-based algorithms. A third simplification (decomposition of NB/WB process, state aggregation) is carried out by the MDP_D and MDP_A algorithms. The CAC and routing algorithms for the exact and decomposed link MDP models are outlined in Section 4 (4) of part I (II) of this paper. One simulation run with the MDP-based algorithms consists of an initial 'warm up' period, followed by a number of adaptation periods, and finally a measurement period. Each adaptation period consists of an measurement period followed by a policy iteration step.

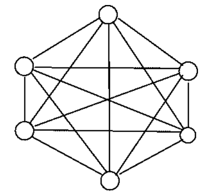
The performance of the least loaded routing (LLR) method is also evaluated. The reason for evaluating LLR is that it is among the routing methods with best performance [17, 19]. The LLR routing method is implemented in many countries, including USA and Canada. We are not aware of any implementation of MDP routing in real networks.

One simulation run with the LLR algorithm consists of one warm up period followed by one measurement period. During these periods, 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.

6.2. *Examples and results*

The performance analysis is performed for the network example W6N described in Table 5. 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 [20] and is shown in Table 6. Network W6N corresponds to a STM type network. The OD pairs in W6N are offered different traffic volumes (asymmetric case). Each OD pair has one direct path and four alternative paths with two links. The algorithm specific parameter settings, presented in Table 7, were determined heuristically based on simulation experience.

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



W6N

Figure 3. Network example W6N.

Table 5. 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

Table 6. 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 7. Algorithm specific parameters.

MDP adaptation epochs	6
MDP_D adaptation epochs	6
MDP_A adaptation epochs	4
Call events in warm up period	500 000
Call events in adaptation period	1000 000
Call events in measurement period	1000 000
Delay penalty weight	100
#Simulation points per curve N	4, 16, 19
#Simulation runs per point M	20
TR parameters (θ_n^n, θ_w^n), $\frac{\text{NB traffic}}{\text{WB traffic}} \leq 1$, $L^s = 0$	(6,0)
TR parameters (θ_n^n, θ_w^n), otherwise	(0,0)
Filtering probability $\phi_{njk}^s(\boldsymbol{x}_n, x_w', \pi)$	1.0
Filtering probability $\phi_{wjk}^s(\boldsymbol{x}_w, \pi)$	1.0

Table 8. 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_D	0.05 (0.0002)	0.00 (0.0000)	0.07 (0.0004)
LLR	0.04 (0.0003)	0.00 (0.0000)	0.17 (0.0184)

Table 9. 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_D	0.13 (0.0288)	0.00 (0.0000)	0.14 (0.0268)
LLR	0.06 (0.0020)	0.00 (0.0000)	0.08 (0.0014)

Table 10. Pooled variance in simulations with variable traffic ratio for $L^s = 0$ case.

	L [%] $s^2(S^2)$
MDP	0.16 (0.0053)
MDP_D	0.23 (0.0392)
LLR	0.15 (0.0154)

Table 11. 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_D	0.03 (0.0003)	0.00 (0.0000)	0.05 (0.0006)
LLR	0.03 (0.0004)	0.00 (0.0000)	0.07 (0.0026)

Table 12. 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_D	0.15 (0.1)	12.0 (609.4)
LLR	0.2 (0.0)	0.0 (0.0)

of the pooled variance, of the performance, in Tables 8–12. We compute the pooled variance over the N simulation points as:

$$s^2 = \frac{1}{N}\sum_{k=1}^{N} s_k^2 \tag{61}$$

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

$$s_k^2 = \frac{1}{M-1}\sum_{i=1}^{M}(x_{ik} - \bar{x}_k)^2 \tag{62}$$

The sample variance of the pooled variance is obtained as:

$$S^2 = \frac{1}{N-1}\sum_{k=1}^{N}(s_k^2 - s^2)^2 \tag{63}$$

Table 13 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 12.

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

$$L = 1 - \bar{R}/R \tag{64}$$

$$\bar{D} = \sum_s \bar{D}_s \frac{\lambda_w^s}{\lambda_w} \tag{65}$$

$$L_D = 1 - \bar{R}_D/R \tag{66}$$

Figures 4 and 5 show the computational time complexity of one cycle of policy iteration for a single link. 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 call queuing.

Table 13. 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_D	148.2	19.9 (157.6)
MDP_A	45.1	0.3 (0.01)
LLR	14.6	10.8 (6.4)

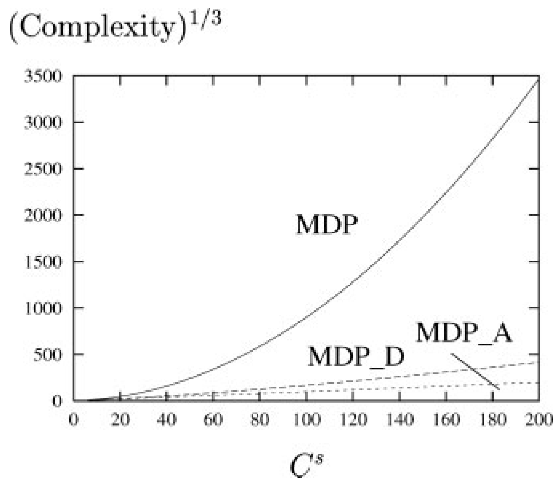


Figure 4. Cube root of the complexity of one policy iteration cycle as a function of the link capacity for the pure loss case.

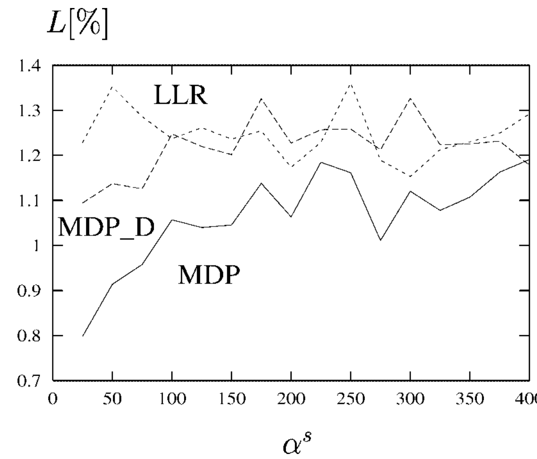


Figure 6. Reward loss of different routing methods versus delay penalty weight.

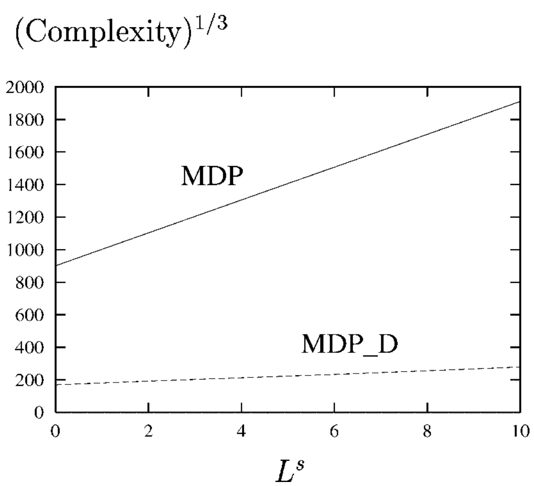


Figure 5. Cube root of the complexity of one policy iteration cycle as a function of the maximal queue size.

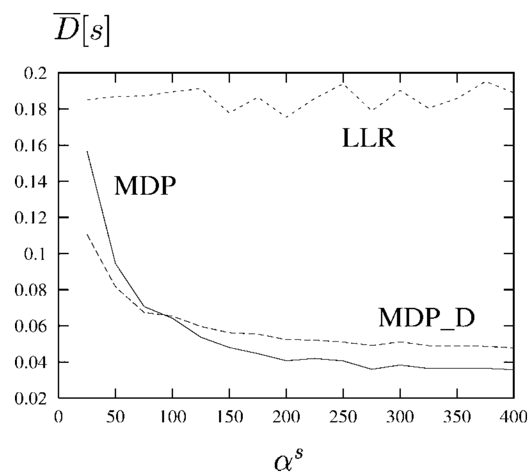


Figure 7. Average call set-up delay of different routing methods versus delay penalty weight.

Figures 6–8 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 9–11 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 12 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 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 13–15 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 16 and 17 show the loss-delay network call blocking probability as a function of the normalised WB reward parameter. The normalised reward parameter, r'_j,

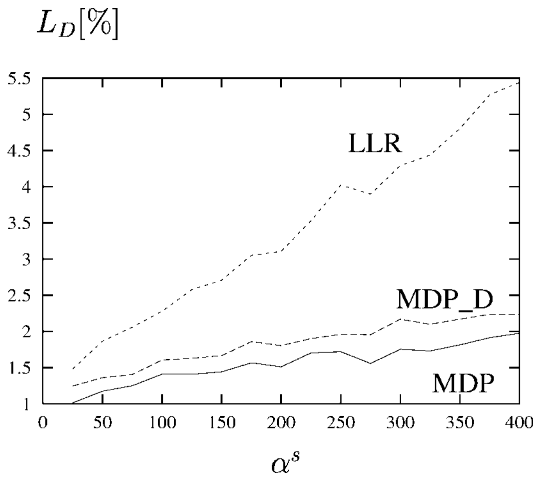


Figure 8. Objective reward loss of different routing methods versus delay penalty weight.

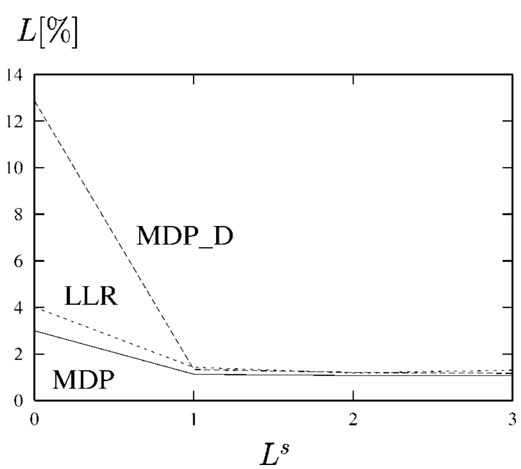


Figure 9. Reward loss of different routing methods versus maximal queue size.

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.

6.3. Results analysis

From the graphs in Figures 4 and 5 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.
- The cube root of the complexity grow linearly as a function of the maximal queue size, both for the exact MDP and decomposed MDP_D link model.

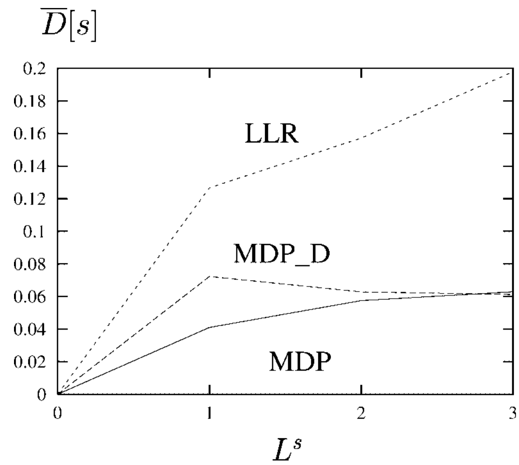


Figure 10. Average call set-up delay of different routing methods versus maximal queue size.

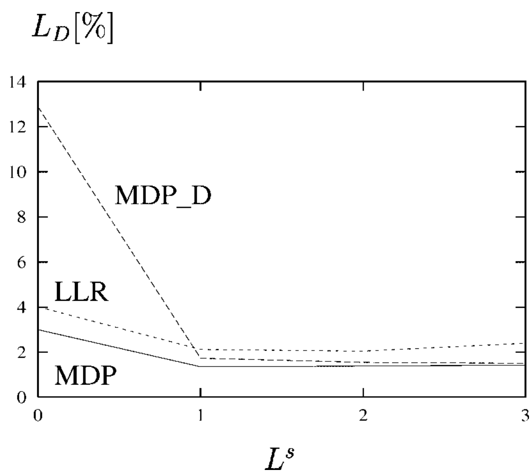


Figure 11. Objective reward loss of different routing methods versus maximal queue size.

From the graphs in Figures 6–8, 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.
- The objective reward loss increases with the delay penalty weight α^s for all the methods.

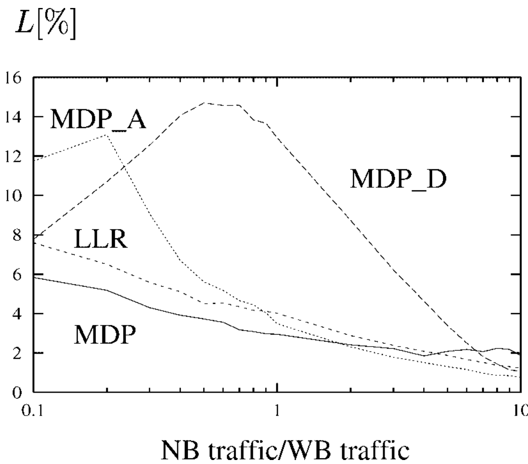


Figure 12. Reward loss of different routing methods versus traffic ratio for $L^s = 0$.

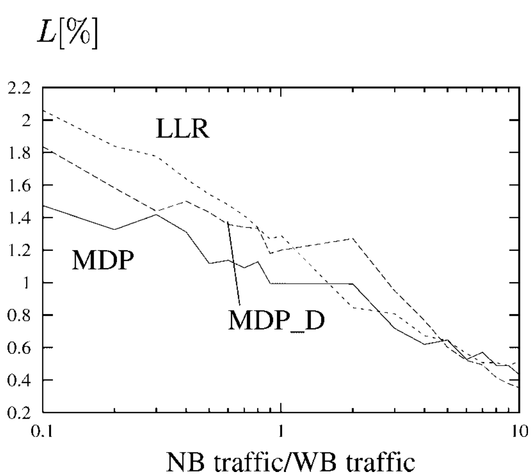


Figure 13. Reward loss of different routing methods versus traffic ratio for $L^s = 3$.

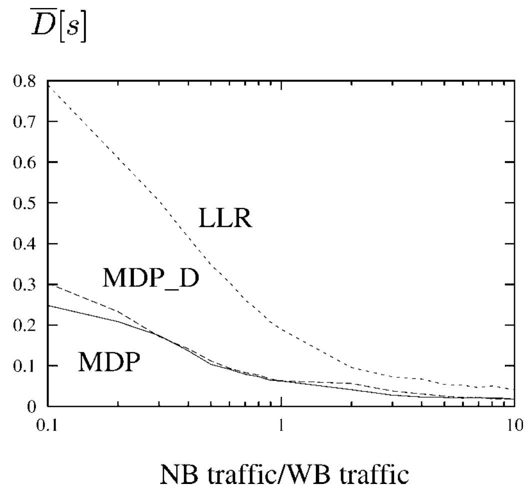


Figure 14. Average call set-up delay of different routing methods versus traffic ratio for $L^s = 3$.

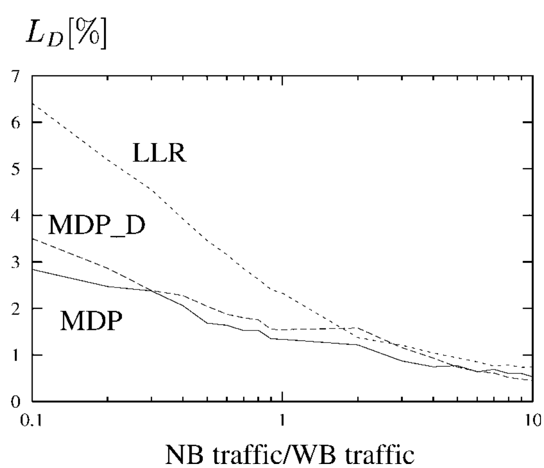


Figure 15. Objective reward loss of different routing methods versus traffic ratio for $L^s = 3$.

From the graphs in Figures 9–11, which shows 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 exact MDP routing algorithm.
- The MDP_D routing algorithm gives high reward loss when $L^s = 0$.

From the graphs in Figure 12, 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.

Since the MDP_D method was not designed for pure loss case, its performance in this case is inferior to the performance of other methods. The reason for that behaviour

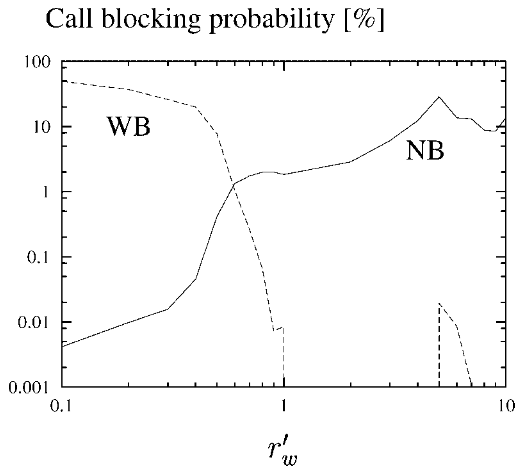


Figure 16. Per-category call blocking probability for MDP method versus normalised reward parameter.

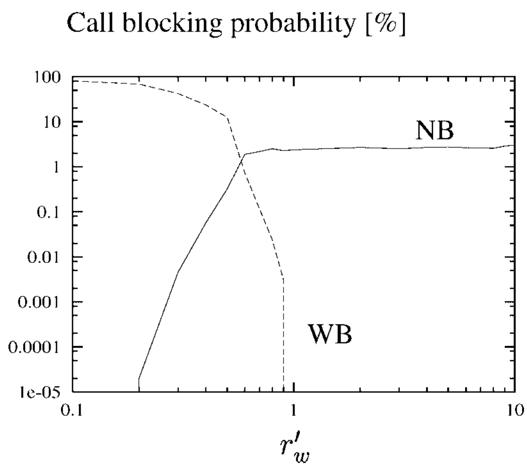


Figure 17. Per-category call blocking probability for MDP_D method versus normalised reward parameter.

is that in the pure loss case the MDP_D model does not have any 'intelligent blocking'.

From the graphs in Figures 13–15, which shows the loss-delay network routing performance versus the traffic ratio when $L^s = 3$, the following conclusions are drawn:

- 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.

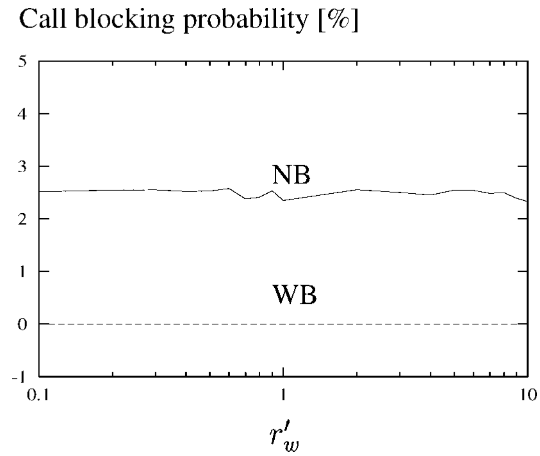


Figure 18. Per-category call blocking probability for LLR method versus normalised reward parameter.

- The best overall behaviour in terms of the objective reward loss is obtained for the MDP-based routing algorithms.

From the graphs in Figures 16–18, which shows 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 method 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.

7. 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 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 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, the low-dimensional link Markov process and link reward process are 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 least loaded routing (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.

The approximate MDP method based on link Markov process decomposition shows good performance for loss-delay networks. However, in pure loss networks, the same method performs poorly in the studied simulation example. In the loss network case, we recommend to use other methods such as state aggregation.

ACKNOWLEDGEMENTS

This work was funded in part by the Swedish Knowledge Foundation and also funded in part by the Natural Sciences and Research Council, Canada (NSERC).

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*. Wiley: Chichester, 1986.
5. Nordström E, Dziong Z. CAC and routing in multi-service networks with blocked wide-band calls delayed, part I: exact link MDP framework. *European Transactions on Telecommunications* 2005; **16**(6): 495–508.
6. 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: Washington DC, USA, 1990; pp. 39–46
7. Krishnan K, Hübner S. Admission control and routing for multirate circuit-switched traffic. In *Proceedings 15th ITC*, Ramaswami V, Wirth PE (eds). Elsevier Science B.V.: Washington DC, USA, 1997; pp. 1043–1055.
8. Marbach P, Mihatsch O, Tsitsiklis J. Call admission control and routing in integrated services networks using neuro-dynamic programming. *IEEE Transactions on Communications* 2000; **18**(2):197–208.
9. Rummukainen H, Virtamo J. Polynomial cost approximations in Markov decision theory based call admission control. *IEEE/ACM Transactions on Networking* 2001; **9**(6):769–779.
10. Kaufman J. Blocking in a shared resource environment. *IEEE Transactions on Communications* 1981; **COM-29**(10):1474–1481.
11. Roberts J. A service system with heterogeneous user requirements. In *Performance of Data Communications Systems and Their Applications*, Pujolle G (ed.). North-Holland, Amsterdam, 1981.
12. Nordström E, Carlström J. A new reward model for MDP state aggregation with application to CAC and routing. *European Transactions on Telecommunications* (in press) 2005.
13. Liao K-Q, Mason LG. An approximate performance model for a multislot integrated services system. *IEEE Transactions on Communications* 1989; **37**(3):211–221.
14. ITU-D SG 2, Teletraffic Engineering Handbook, Available at http://www.tele.dtu.dk/teletraffic, 2005.
15. Dziong Z, Mignault J, Rosenberg C. Blocking evaluation for networks with reward maximization routing. In *Proceedings of INFOCOM'93*, San Fransisco, USA, 1993.
16. 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.
17. Dziong Z. *ATM Network Resource Management*. McGraw-Hill: Columbus OH, USA (ISBN 0-07-018546-8), 1997.
18. Dziong Z, Mason LG. Call admission and routing in multi-service loss networks. *IEEE Transactions on Communications* 1994; **42**(2):2011–2022.
19. Ash G. *Dynamic Routing in Telecommunications Networks*. McGraw-Hill: New York, 1998.
20. 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*'.

