Call Admission Control and Routing for Integrated CBR/VBR and ABR Services: A Markov Decision Approach

Ernst Nordström and Jakob Carlström

Department of Computer Systems, Information Technology, Uppsala University
Box 325, SE–751 05 Uppsala, Sweden
Tel: +46 18 471 31 11
Fax: +46 18 55 02 25
E-mail: {ernstn,jakobc}@DoCS.UU.SE

Abstract

In this paper we study the Call Admission Control (CAC) and routing issue for ATM networks which carry integrated CBR/VBR and ABR traffic. The integration of CBR/VBR and ABR traffic is assumed to be based on the max-min fairness criterion. The CAC and routing task is formulated as a Markov decision problem (MDP) where the objective is to maximize the revenue from carried calls. The MDP is solved by the policy iteration procedure, after applying the standard link independence assumption. The numerical results show that the MDP routing method yields higher revenue than the least loaded path routing method.

1 Introduction

In an ATM network supporting Constant Bit Rate (CBR), Variable Bit Rate (VBR) and Available Bit Rate (ABR) services the resource utilization will be controlled by Call Admission Control (CAC) and routing at call set up time and by flow control during the data transfer phase.

The ATM Forum identifies max-min fairness as one of the main design goals for integration of CBR/VBR and ABR services [1]. This means that on each link, the ABR calls are offered a fair share of the free bandwidth not used by the CBR/VBR calls. If an ABR call can not use up its fair share because it has lower source rate or is assigned a lower bandwidth on another link, the excess bandwidth is split fairly among all other ABR calls. Several flow control algorithms that give max-min fairness have been proposed, including binary feedback schemes [2] and explicit rate feedback schemes [3]. A routing scheme for the ABR service environment based on max-min fairness was proposed in [4].

The objective of CAC and routing in ATM networks with on-demand call set-up is maximization of the network revenue and maintenance of the service availability for network customers. In this paper, we propose a CAC and routing method for on-demand set-up of CBR/VBR and ABR calls that is based on the max-min fairness criterion and the Markov decision problem (MDP) concept. The MDP concept has been successfully applied to CAC and routing in ATM networks with on-demand set-up of CBR/VBR calls [5] and with delayed set-up of CBR/VBR calls [6, 7, 8]. The MDP computations used in this paper are based on policy iteration, which is a form of dynamic programming [9].

We assume that the ABR flow control algorithm used by the network is able to achieve max-min fairness. The routing algorithm models the max-min fair bandwidth allocation by a simple iterative formula [4]. The ABR calls sharing a link may use different shares of the link bandwidth since different calls may have their bandwidth constrained by different bottleneck links.

The numerical section shows that the revenue performance of the MDP method is better than the least loaded path (LLP) routing method. A major advantage of the MDP method is also shown: the ability to control the per-class call blocking probabilities as a function of per-class reward parameters.

The paper is organized as follows. In the next section, the routing problem is introduced. Section 3 describes how to compute the max-min fair bandwidth share for ABR calls. Section 4 presents a decomposed Markov decision model for the CAC and routing task. Section 5 describes the computations needed for solving the decomposed Markov decision problem. Section 6 presents the numerical results and section 7 concludes the paper.

2 The routing problem

An ATM network can be described as a set of nodes and a set of links connecting the nodes. The ATM network studied in this paper is offered calls from K classes. The classes are of two types: ABR (type 1) and CBR/VBR (type 2). The j-th class is characterized by an origin-destination (OD) node pair, peak bandwidth requirement (for the CBR and ABR classes) or equivalent bandwidth requirement (for the VBR class), b_j, rate of arrival process (assumed to be Poissonian), λ_j, mean holding time, $1/\mu_j$, (the holding time is assumed to be exponentially distributed), set of alternative routes, W_j, and a reward parameter $r_j \in (0, \infty)$ which is the average reward for carrying the j-th class call. A class of an ABR type is characterized by the minimum mean holding time, since the actual mean holding time is state dependent. A minimal accepted service ratio, ϱ_{min}, is also specified for the ABR classes. On-demand call set-up is assumed, i.e. calls are not allowed to wait in a queue when the network is busy.

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

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

where $\bar{\lambda}_j$ denotes the average rate of accepted class-j calls.

3 Max-min fair share computations

In this section we describe how to model the call-level max-min fair bandwidth allocation, which should be considered by the routing method. Note that the service ratio for an ABR call is its fair share divided by the peak bandwidth requirement.

A call is said to be saturated if it has reached its desired peak rate or a link on the route traversed by the call is saturated. A link is said to be saturated if all of its bandwidth has been allocated to calls sharing the link. Let G be the set of all ABR calls in the network, G_s the set of calls using link s, and *sat* and *unsat* the set of saturated and unsaturated calls, respectively. For link s, let sat_s be the set $G_s \cap sat$, and $unsat_s$ be the set $G_s \cap unsat$. Given a set H of calls, let $L(H)$ be the set of all links in the network with at least one call in H using them. Let C_s be the capacity of link s and let x_2 denotes the number of CBR/VBR calls present on link s. The algorithm for computing the max-min fair shares can be described as follows [4]:

1. Initialization: $sat = \emptyset$ and $unsat = G$.
2. Iteration: Repeat the following steps until *unsat* becomes empty.

- For every link $s \in L = L(unsat)$, calculate

$$share_s = \frac{C_s - x_2 b_2 - \sum_{i \in sat_s} b_{1i}}{|unsat_s|}$$

- Get the minimum: $min\_share = \min\{share_s \mid s \in L\}$.
- Update rate b_{1i}, $i \in unsat_s$, $s \in L$: $b_{1i} = min\_share$.
- Move new saturated calls in *unsat* to *sat*.

4 Markov decision model

To obtain a feasible computational complexity when solving the MDP, the network is decomposed into a set of links, assumed to have independent traffic and reward processes [10]. The link states are assumed to be statistically independent and link arrivals are state-dependent Poisson streams. In particular, these assumptions imply that a call connected to a path consisting of l links is decomposed into l independent calls with the same mean holding time as the original call.

In the decomposed Markov decision model, the state of each link can be described by a vector $x = \{x_j\}$, where x_j denotes the number of class-j calls accepted on link s. To simplify the notation, we assume that alternative routes for class j have no common links — this is not a limitation of the approach. The state space is given by:

$$X = \left\{ x : x_j \geq 0, \sum_{j \in J_1} x_j \varrho_{min} b_j + \sum_{j \in J_2} x_j b_j \leq C_s \right\} \tag{2}$$

The SMDP action a for each link is represented by a vector $a = \{a_j\}$ corresponding to admission decisions for presumptive call requests. The action space A is a given by:

$$A = \left\{ a = \{a_j\} : a_j \in \{0, 1\}, \; j \in J \right\} \tag{3}$$

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

$$A(x) = \left\{ a \in A : a_j = 0 \text{ if } x + \delta_j \notin X,\ j \in J \right\} \tag{4}$$

where δ_j denotes a vector of zeros except for a one at position j.

The state transition probabilities for each link can be written as:

$$P_{xy} = \begin{cases} \lambda_{sj}(x,\pi) a_j \tau(x,a), & y = x + a_j \delta_j \in X,\ j \in J \\ x_j \mu_j \bar{\varrho}_s(x) \tau(x,a), & y = x - \delta_j \in X,\ j \in J_1 \\ x_j \mu_j \tau(x,a), & y = x - \delta_j \in X,\ j \in J_2 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where $\lambda_{sj}(x,\pi)$ denotes the arrival rate of class j calls to link s in state x under routing policy π, $\bar{\varrho}_s(x)$ denotes the average service ratio in state x, and $\tau(x,a)$ denotes the average sojourn time in state x. The call arrival rate $\lambda_{sj}(x,\pi)$ is given by:

$$\lambda_{sj}(x,\pi) = \lambda_{jk}(\pi)\ \phi_{sj}(x,\pi) \prod_{c \in k \setminus \{s\}} (1 - B_{cj}(\pi)) \tag{6}$$

where $B_{cj}(\pi)$ denotes the probability that link c has not enough capacity to accept a class j call, $\phi_{sj}(x,\pi)$ denotes a filtering probability that the route net-gain is positive. This probability can be computed from link state distributions, or approximated with one according to experiments in [5]. The $\lambda_{jk}(\pi)$ denote the arrival rate of class j to route $k \in W_j$ and is given by a load sharing model [5]:

$$\lambda_{jk}(\pi) = \lambda_j \frac{\bar{\lambda}_{jk}(\pi)}{\sum_{h \in W_j} \bar{\lambda}_{jh}(\pi)} \tag{7}$$

where $\bar{\lambda}_{jk}(\pi)$ denotes the rate of accepted class j calls to route k, and λ_j denotes the arrival rate of class j.

The average service ratio $\bar{\varrho}_s(x)$ can be estimated by the partial blocking service ratio [11]:

$$\bar{\varrho}_s(x) \approx \begin{cases} 1, & \sum_{j \in J} x_j b_j \le C_s \\ \left[C_s - \sum_{j \in J_2} x_j b_j \right] \Big/ \left[\sum_{j \in J_1} x_j b_j \right], & \sum_{j \in J} x_j b_j > C_s \end{cases} \tag{8}$$

The ABR calls on a saturated link can have different service ratios, but their average is always given by the partial blocking service ratio. The formula for the partial blocking service ratio is also valid when all the ABR calls on the link have unit service ratio. However, for an unsaturated link, the partial blocking service ratio will overestimate the average service ratio.

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

$$\tau(x,a) = \left\{ \sum_{j \in J_1} x_j \bar{\varrho}_s(x) \mu_j + \sum_{j \in J_2} x_j \mu_j + \sum_{j \in J} a_j \lambda_{sj}(x,\pi) \right\}^{-1} \tag{9}$$

The expected immediate reward in state x is given by $R(x,a) = q(x)\tau(x,a)$, where $q(x)$ is given by:

$$q(x) = \sum_{j \in J_1} r_{sj}(\pi) x_j \bar{\varrho}_s(x) \mu_j + \sum_{j \in J_2} r_{sj}(\pi) x_j \mu_j \tag{10}$$

where $r_{sj}(\pi)$ is the class-j link reward parameter which fulfils the obvious constraint:

$$r_j = \sum_{s \in k} r_{sj}(\pi) \tag{11}$$

The numerical results presented in [5] indicate that a simple division rules yield sufficient control performance, i.e. $r_{sj}(\pi) = r_j / l$, where l equals the number of links in the route.

Even in the decomposed model, the state space can be quite large if many call classes share the links. One way to reduce the state space is to construct a modified reward process in which the link call classes with the same bandwidth requirement and mean holding time are aggregated into one class i with average reward parameter defined as [5]:

$$r_{si}(\pi) = \frac{\sum_{j \in i} r_{sj}(\pi)\bar{\lambda}_{sj}(\pi)}{\sum_{j \in i} \bar{\lambda}_{sj}(\pi)} \tag{12}$$

where $\bar{\lambda}_{sj}(\pi)$ denotes the average rate of class-j calls accepted on link s. This simplification reduces the number of effective classes to the number of classes with unique bandwidth requirement and holding time.

5 Markov decision computations

When a call request arrives to the network, the *gain* of accepting the call on each route connecting the call's OD pair is computed and the route with the largest positive gain is chosen to carry the new call. The call request is rejected if the largest gain is negative. The gain simply measures the increase in long-term reward due to call acceptance on the given route. The gain for route k for class j call can be expressed in terms of the call's reward parameter r_j and the link shadow prices, $p_{sj}(x,\pi)$, $s \in k$, paid for establishing the call on each links s in state x under routing policy π [5]:

$$g_k = r_j - \sum_{s \in k} p_{sj}(x, \pi) \tag{13}$$

The link shadow prices are computed using link reward parameters, $r_{sj}(\pi)$, and link gains, $g_{sj}(x,\pi)$:

$$p_{sj}(x, \pi) = r_{sj}(\pi) - g_{sj}(x, \pi) \tag{14}$$

The link gains $g_{sj}(x,\pi)$ can be expressed in terms of the relative value function $v_s(x,\pi)$:

$$g_{sj}(x, \pi) = v_s(x + \delta_j, \pi) - v_s(x, \pi) \tag{15}$$

The difference $v_s(x,\pi) - v_s(y,\pi)$ can be interpreted as the expected difference in accumulated reward computed over an infinite interval starting in state x instead of in state y under routing policy π.

The optimal routing policy and the relative value function for all links can be determined by policy iteration, which is a form of dynamic programming. The policy iteration is performed for each link and involves three steps:

- task identification,
- value determination,
- policy improvement.

The first step involves determining the Markov decision model, i.e. state transition probabilities $p_{xy}(a)$ and expected immediate rewards $R(x,a)$.

The second step involves computing the relative value function for the current policy. The value determination step for link s consists of solving the set of linear equations:

$$\begin{cases} v_s(x, \pi) = R(x, a) - g_s(\pi)\tau(x, a) + \sum_{y \in X} p_{xy}(a) v_s(y, \pi), & x \in X \\ v_s(x_r, \pi) = 0, & x_r \in X \end{cases} \tag{16}$$

where x_r is an arbitrary chosen reference state (e.g. the empty state) and $g_s(\pi)$ denotes the average reward rate. The solution involving all the $v_s(x,\pi)$ and $g_s(\pi)$ can be obtained by any standard routine for sparse linear systems.

The third step is the actual policy improvement. This step consists of finding, for each link s, the action that maximizes the relative value in each state:

$$a = \operatorname*{argmax}_{u \in A(x)} \left\{ R(x, u) - g_s(\pi)\tau(x, u) + \sum_{y \in X} p_{xy}(u) v_s(y, \pi) \right\}, \quad x \in X \tag{17}$$

Policy iteration can be proved to converge to an optimal policy in a finite number of iterations in the case of finite state and action space [9].

The proposed method can be summarized as follows:

1. Choose initial link admission control policies and relative functions $v_s(x,\pi)$ for all the links.
2. During a finite period, allocate calls according to the route gain functions obtained from the relative value functions. At the same time, measure traffic statistics (call arrival rates, per-route call acceptance rates, call departure rates) in order to identify the Markov decision tasks for the current link admission control policies.

3. Evaluate the applied policies in the context of the current Markov decision tasks, by solving sparse linear equation systems, and improve the policies for all the links.
4. Repeat steps 2 and 3 until the average reward rate converges.

6 Numerical results

This section presents results on routing performance for three different routing methods:

- MDP — Markov decision routing based on the link independence assumption,
- MDP_P — Markov decision routing based on the link independence assumption, with priority for the direct path,
- LLP — Least loaded path routing.

In the MDP_P routing method the direct path is tried first, and if its route gain is negative, a multi-link path is chosen according to the maximum path net-gain rule.

In LLP routing, a new call is offered to an admissible route which has the least loaded bottleneck link. We use two different measures of the link load: the allocated bandwidth and the average service ratio. The average service ratio is only used if the bottleneck links of all the routes have no free bandwidth. Note that we use the same procedure to compute the bandwidth shares for the ABR calls as in Markov decision routing.

The performance analysis was performed for the W13N network example, which models a typical ATM network of low connectivity (see Table1 and Figure 1). Each curve in the figures is based on 30 simulations runs, and 95% confidence intervals are shown for each curve.

#nodes	13
μ_j^{-1}	1, 10
b_j	1, 6
WB CBR traffic	38%
traffic [Erlang]	527
link capacity	50, 100
$r_j' = r_j \mu_j / b_j$	1
max #links in path	3

Table 1: Description of network W13N.

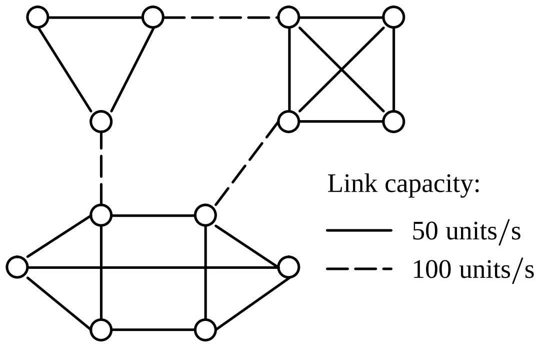

Figure 1: Topology of network W13N.

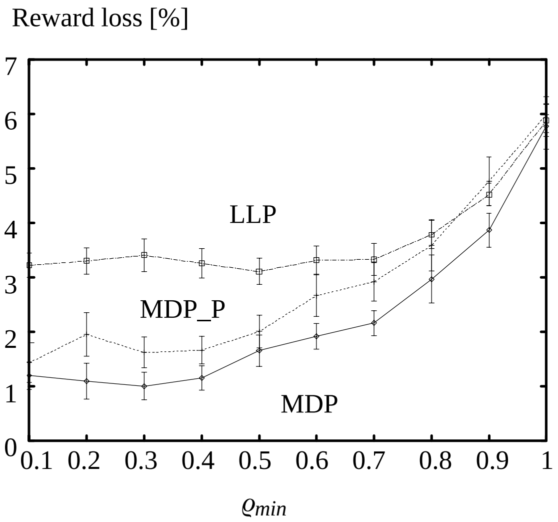

Figure 2: Reward loss as a function of minimal service ratio for different routing methods.

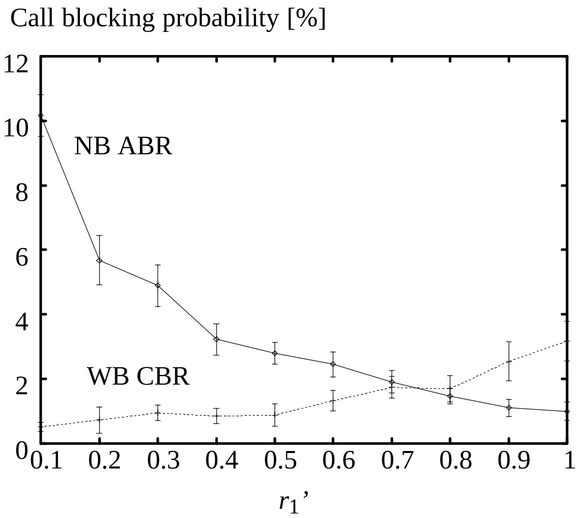

Figure 3: Per-class call blocking probabilities as function of the normalized reward parameter for the narrow-band ABR class for the MDP routing method with $\varrho_{min} = 0.5$.

Figure 2 shows the overall reward loss as a function of the minimal service ratio for the three different routing methods. The overall reward loss is computed as $L = 1 - \overline{R}/R$, where $\overline{R}$ denotes the reward actually received, and R denotes the potential reward. Both variants of the MDP method yields lower reward loss than the LLP method. In contrast to routing in pure CBR/VBR networks [5], the MDP method with priority for direct path yields a slightly higher reward loss than the normal MDP method. This is explained by the fact that the priority for direct path tends to reduce the average service ratio of the links, which reduces the average reward rate.

Figure 3 shows the call blocking probability for the narrow-band ABR class and the wide-band CBR class for different values of the normalized reward parameter, $r_j' = r_j\mu_j/b_j$, for the ABR class. The results were obtained by the normal MDP method for a minimum service ratio ϱ_{min} of 0.5. The figure shows an important feature of the MDP routing method: the ability to control the per-class call blocking probabilities as a function of normalized reward parameters.

7 Conclusion

In this paper we studied the CAC and routing problem for ATM networks which carry integrated CBR/VBR and ABR traffic. The integration of CBR/VBR and ABR traffic is assumed to be based on the max-min fairness criterion. The CAC and routing task was formulated as a Markov decision problem (MDP) with the objective to maximize the revenue from carried calls. To obtain a feasible computational complexity, the network was decomposed into a set of links with independent traffic and reward processes. The MDP routing policy was computed by the policy iteration algorithm.

The numerical results showed that the MDP routing method yields higher average revenue rate than the least loaded path routing method. The results also showed an important feature of the MDP method: the ability to control the the per-class call blocking probabilities as a function of normalized reward parameters.

In our future work we will study different models of the per-link average service ratio for ABR calls and perform numerical simulations for different network topologies.

Acknowledgements

This work was financially supported by NUTEK, the Swedish National Board for Industrial and Technical Development.

REFERENCES

[1] J. Jaffe, “Bottleneck Flow Control”, *IEEE Trans. on Commun.*, vol. COM-29, no. 7, 1981.

[2] M. Hluchyj and N. Yin, “On Closed-Loop Rate Control for ATM Networks”, *in proc. of INFOCOM’94*, 1994.

[3] A. Charny, D. Clark and R. Jain, “Congestion Control with Explicit Rate Indication”, *in proc. International Conference on Communications, ICC’95*, 1985.

[4] Q. Ma, P. Steenkiste and H. Zhang, “Routing High-Bandwidth Traffic in Max-min Fair Share Networks”, *in proc. of SIGCOMM’96*, CA, USA, 1996.

[5] Z. Dziong and L. Mason, “Call Admission and Routing in Multi-Service Loss Networks”, *IEEE Trans. on Commun.*, vol. 42, no. 2, 1994.

[6] Z. Dziong, K. Liao and L. Mason, “Flow Control Methods in Multi-Service Networks with Delayed Call Set Up”, *in proc. of INFOCOM’90*, pp. 39-46, San Francisco, USA, 1990.

[7] Z. Dziong and E. Nordström, “Call Admission and Routing for Broadband Networks with Delayed Set Up of Wideband Calls — Markov Decision Theory Framework”, *submitted*, 1999.

[8] E. Nordström, “Near-Optimal Link Allocation of Blockable Narrow-Band and Queueable Wide-Band Call Traffic in ATM networks”, *in proc. of the 15th International Teletraffic Congress, ITC’15*, Washington D.C., USA, 1997.

[9] H. Tijms, *Stochastic Modeling and Analysis — a Computational Approach*, Wiley, 1986.

[10] E. Nordström, *Markov Decision Problems in ATM Traffic Control*, PhD thesis DoCS 98/100, Department of Computer Systems, Uppsala University, Sweden, 1998.

[11] S. Blaabjerg and G. Fodor, “A Generalization of the Multirate Circuit Switched Loss Model to Model ABR Services in ATM Networks”, *in proc. of the IEEE International Conference on Communications*, Singapore, 1996.