A Reinforcement Learning Scheme for Adaptive Link Allocation in ATM Networks

Ernst Nordström, Jakob Carlström

Department of Computer Systems, Uppsala University,
Box 325, S–751 05 Uppsala, Sweden
Fax: +46 18 55 02 25,
<http://www.docs.uu.se/docs/ann/>
jakobc@DoCS.UU.SE, ernstn@DoCS.UU.SE

Abstract

This paper presents an adaptive scheme for a sub-function in Asynchronous Transfer Mode (ATM) network routing, called link allocation. The scheme adapts the link allocation policy to the offered Poisson call traffic such that the long-term revenue in maximized. It decomposes the link allocation task into a set of link admission control (LAC) tasks, formulated as semi-Markov Decision Problems (SMDPs). The LAC policies are directly adapted by reinforcement learning. Simulations show that the direct adaptive SMDP scheme outperforms static methods, which maximize the short-term revenue. It also yields a long-term revenue comparable to an indirect adaptive SMDP method.

1 Introduction

Routing in public Asynchronous Transfer Mode (ATM) networks has two objectives: maximizing the operator revenue and maintaining the network availability for different call types. Adaptive routing techniques are efficient when the traffic demand varies over time. The approach presented in [1], views the routing task as an adaptive semi-Markov Decision Problem (SMDP). The method selects a route from a set of candidate routes, the objective being to maximize the long-term revenue. It uses an indirect algorithm, which adapts a model of the underlying controlled Markov Process, and computes control policies based on the latest model. In order to simplify the revenue analysis, the call traffic load and revenue generation on successive transmission links are assumed to be independent.

In this paper, we assume that two adjacent switches are interconnected by a set of parallel transmission links. The adaptive routing problem is decomposed into a set of adaptive link allocation problems,where the task is to select the link within a link group that maximizes the long term revenue. An adaptive link allocation scheme, based on a direct (model-free) SMDP approach is presented. A near-optimal link allocation policy is found by solving a series of simple link admission control (LAC) tasks, formulated as direct SMDPs. The link admission controllers use reinforcement learning [2] [3], in form of the actor-critic method [4], to find optimal state-dependent LAC policies. In particular, the controllers should detect link states where blocking of narrow-band calls leads to higher long-term revenue. A set of functions that measure the relative merit of accepting a call in a particular link state, controls the link allocation after adaptation.

The experimental results show that the proposed scheme has comparable performance with the indirect adaptive SMDP method, both in terms of long-term revenue and in terms of adaptation rate.

2 The Link Allocation Problem

In the link allocation problem, a group of M links with capacities C_i [units/s], $i \in I = \{1,..., M\}$, is offered calls from K different classes. Calls belonging to a class $j \in J = \{1,..., K\}$ have the same bandwidth requirements b_j [units/s], and similar arrival and holding time dynamics. As in [1], we assume that type-j calls arrive according to a Poisson process with intensity λ_j [s^{-1}], and that the call holding time is exponentially distributed with mean $1/\mu_j$ [s]. In this work, the parameter b_j is given by the peak ATM cell transmission rate, since deterministic cell multiplexing is assumed.

The task is to find a link allocation policy π that maps *request states* $(j,n) \in J \times N$ to *allocation actions* $a \in A$, $\pi: J \times N \rightarrow A$, such that the long-term revenue is maximized. The set N contains all feasible link group states, and the set A contains the possible allocation actions, $I \cup \{REJECT\}$. The set of feasible link group states is given by the Cartesian product of the sets of feasible link states N_i,

$$N_i = \left\{ n_i : n_{ij} \geq 0,\ j \in J; \sum_{j \in J} n_{ij} b_j \leq C_i \right\},\ i \in I\ ,$$

where n_{ij} is the number of type-j calls accepted on link i.

The network availability constraint (limited call blocking probabilities) is not considered in the present work. Moreover, we assume an uniform call charging policy, which means that the long-term revenue is proportional to the cell throughput at the call level.

3 An Adaptive Link Allocation Scheme

In order to speed up the adaptation process, the link allocation task is decomposed into a set of link admission control (LAC) tasks with actions $a_i \in A_i = \{ACCEPT,$

REJECT}, see Figure 1. The link admission controllers adapt to a constant-rate call flow, during a number of periods. The call flows are kept unchanged during each period, which ends when an optimal LAC policy has been found for each link. Then, new call flows are determined for the following period, based on the performance of the LAC policies determined during the previous period.

A load sharing link allocation policy with constant load sharing coefficients h_{ij} maintains the LAC task during the policy adaptation period. That is, a type-j call is offered to link i with probability h_{ij} (Figure 1). The selected link admission controller can then accept or reject the call. The load sharing coefficients used during period p are determined by:

$$h_{ij,p} = \frac{\bar{\lambda}_{ij,p-1}}{\sum_{k \in I} \bar{\lambda}_{kj,p-1}}, \quad i \in I, \ j \in J, \tag{1}$$

where $\bar{\lambda}_{ij,p-1}$ denotes the measured rate of accepted type-j calls on link i during period $p-1$. Hence, a link which has a relatively high admission rate will be offered more calls during the next adaptation period. The adaptation stops when the new coefficients $\{h_{ij,p}\}$ are sufficiently close to the old coefficients $\{h_{ij,p-1}\}$.

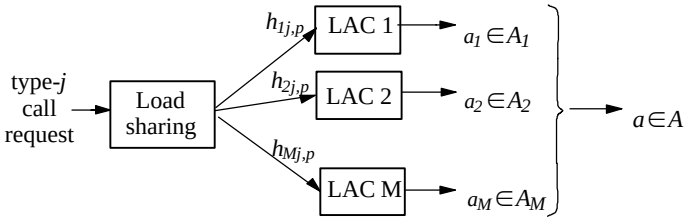

Figure 1: Link allocation during adaptation.

In the course of LAC adaptation, each link admission controller i estimates merit functions $m_{ACCEPT,i}(j,n_i)$, which measure the relative merit of accepting a type-j call in link state n_i. The accept merit functions control the link selection after the adaptation phase.

When a type-j call request arrives, each link is checked to see if it has sufficient free capacity to accept the call. Provided this is the case, the controller selects an action $a_i \in A_i$, with higher probability for the action which yields higher long-term revenue (see section 4). The controller outputs the resulting action a_i along with the accept merit value $m_{ACCEPT,i}$. The link allocator then selects the link with the highest accept merit value (among the links that accept the call), see Figure 2. If all a_i = *REJECT*, the link allocator rejects the call.

In certain link states, called ”intelligent blocking” link states, rejecting calls of some types yields a higher long term revenue than accepting them. They typically

occur when the link has a free link capacity that is equal to the size of a wide-band call. By rejecting a narrow-band call request, the controller reserves bandwidth to the wide-band class, and so increasing the long-term revenue. However, if many narrow-band calls are accepted on the link, at least one of them is likely to depart before the next wide-band call arrives. Hence, narrow-band calls can be accepted, although the free capacity equals the size of a wide-band call.

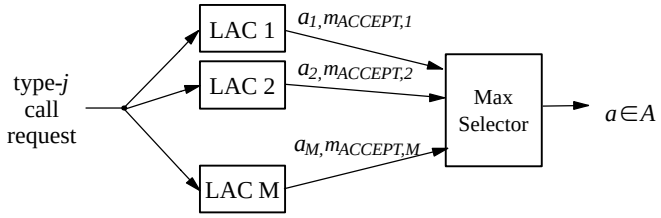

Figure 2: Link allocation after adaptation.

4 Reinforcement learning of the LAC policy

Within each link i, a link admission controller constructs a policy π_i: $X_i \rightarrow A_i$, $A_i = \{ACCEPT, REJECT\}$. $\pi_i(x_i)$ indicates what action to $a_i \in A_i$ to select at each SMDP state $x_i \in X_i$. X_i is defined by $X_i = N_i \times E \times J$, where the two possible types of events, an arrival or a departure of a call, are the elements in $E = \{ARRIVAL, DEPARTURE\}$.

The objective of the link admission controller of link i is to find a policy π_i which maximizes the long-term revenue, expressed as the expected (infinite horizon) discounted reward. This "utility" is denoted $V_{i\pi}(\xi_i)$, for a SMDP state $\xi_i \in X_i$:

$$V_{i\pi}(\xi_i) = E\left\{ \int_{t=0}^{\infty} e^{-\beta t} r_i(x_i(t), a_i(t)) dt \right\} \qquad (2)$$

where the reward $r_i(x_i(t),a_i(t))$ is the continuous-time total cell transmission rate on the link, $x_i(t)$ and $a_i(t)$ denote the SMDP state and action at time t, respectively, and $x_i(0) = \xi_i$. This maximization is performed by a delayed reinforcement learning method, which is a modification of the actor-critic method [4], with its redefinition for SMDPs [3].

The actor-critic method solves the task using two separate function approximators (Figure 3): an evaluation function $V_i(x)$ which models $V_{i\pi}(x)$ and a policy function $\pi_i(x)$. In our modification, π_i is divided into two sub-policies: an arrival policy π_{ia}, which is adaptive, and a departure policy π_{id}, which is deterministic. A sub-policy selector chooses what sub-policy to employ, according to

$$\pi_i(n_i, e, j) = \begin{cases} \pi_{ia}(n_i, j), & e = ARRIVAL \\ \pi_{id}(n_i, j), & e = DEPARTURE \end{cases}, \text{ where} \tag{3}$$

$$\pi_{ia}(n_i, j) \in \{ACCEPT,\ REJECT\}\ , \tag{4}$$

$$\pi_{id}(n_i, j) \equiv ACCEPT\ . \tag{5}$$

The motivation for Equation 5 is that the link admission controller must accept all call departure requests.

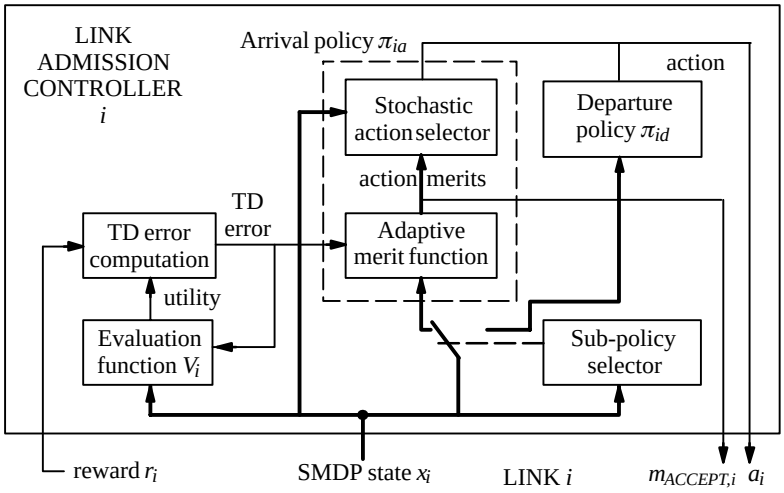

Figure 3: The architecture of the modified actor-critic method

π_{ia} uses an adaptive merit function (Figure 3), which indicates the relative merits $m_{ACCEPT,i}$ and $m_{REJECT,i}$, for accepting or rejecting a requested call, respectively. The accept merits $m_{ACCEPT,i}$ are also output to the link allocation algorithm. A stochastic action selector chooses among the actions, with higher probability for actions with higher merits. The probability of selecting an action a_i in state x_i is determined by the action merits and the SMDP state, by choosing action $a_i(x)$ as in [4]:

$$a_i(x) = \arg\max_{u \in A_i} \left(m_{u,i}(x_i) + e_u \right) \tag{6}$$

where $m_{u,i}(x_i)$ is the merit of action u, and e_u are independent random numbers, drawn from an exponential distribution with mean $1/T(x_i,u_i)$. The "temperature" $T(x_i,u_i)$ adjusts the randomness of action selection. After adaptation, $T(x_i,u_i)$ is set to zero for all (x_i,u_i).

The discounted cumulative reward $q_{i,xy}$ received between two state transitions, from a SMDP state x entered at time t_x, to another SMDP state y entered at time t_y, is defined by

$$q_{i,xy} = \int_{t_x}^{t_y} e^{\beta(t_x - t)} r_i(t - t_x) dt \tag{7}$$

The link admission controller learns from interacting with the link in repeated trials. By definition of the evaluation function and (Equation 2), the desired evaluation function $V_{i\pi}(x)$ must satisfy

$$V_{i\pi}(x) = q_{i,xy} + e^{\beta(t_x - t_y)} V_{i\pi}(y) \tag{8}$$

During learning, this may not be true. The difference between the two sides of the equation is called the temporal difference (TD) error. This is used to update both $V_i(x)$, according to the TD(0) rule [2], and $\pi_i(x)$:

$$\Delta V_i(x) = \eta_V [q_{i,xy} + e^{\beta(t_x - t_y)} V_i(y) - V_i(x)] \tag{9}$$

$$\Delta m_{u,i}(x) = \eta_\pi [q_{i,xy} + e^{\beta(t_x - t_y)} V_i(y) - V_i(x)] \tag{10}$$

where η_V and η_π are "learning rate" parameters, and $u \in A_i$ is the action chosen in state x.

It should be noted that although an effect of using a deterministic departure policy is that the policy is not updated after call departures, the evaluation function is updated, which leads to better estimates of V_i, resulting in faster and safer convergence of the arrival policy. The non-zero probability of choosing and evaluating actions with low merits (Equation 6), allows the link admission controller to improve its policy.

In reinforcement learning, neural networks, for example multi-layer perceptrons, are often used to approximate the evaluation and policy functions. This is beneficial when the state space is too large to explore completely, since the neural network allows generalization between states. Neural networks also allow incorporation of other environment parameters, providing the link admission controller with information which may improve its performance, for example in cases where the Poisson call model does not hold. However, in this work, lookup tables were used for function approximation.

5 Results

The proposed adaptive link allocation scheme was tested on simulated Poisson call traffic. Results for three other methods are presented for comparison: the indirect adaptive SMDP method [1] and the static First Fit and Best Fit methods. The static methods maximize the short-term revenue, using the following algorithms:

- First Fit: Search the links in a predefined order, and allocate the call to the first link found with sufficient capacity.
- Best Fit: Choose the link with least, but sufficient, capacity.

The simulations were done for a link group of 3 links with capacities $C_i = C = 24$ [units/s] for all *i*. The link group was offered calls from two classes, characterized by bandwidth requirements $b_1 = 1$, $b_2 = 6$ [units/s] and call holding times $1/\mu_1 = 1/\mu_2 = 1$ [s]. The arrival intensities λ_1 *and* λ_2 [s^{-1}] were varied so that:

$$\frac{b_1\lambda_1}{C\mu_1} + \frac{b_2\lambda_2}{C\mu_2} = 1.5 \tag{11}$$

The temperature $T(x_i,u)$ of the actor/critic-method was set using prior knowledge of the intelligent blocking states, introduced in section 3. In particular, intelligent blocking should be possible for the narrow-band class, at link states where the free capacity equals the size of one wide-band call, that is, for the link states $n_i \in \{(0,3), (6,2), (12,1)\}$. In the corresponding SMDP states x_i, different temperatures were used for accept and reject actions: $T(x_i,ACCEPT) = 0.4$, and $T(x_i,REJECT) = 0.3$. For all other $(x_i,u) \in X_i \times A_i$, the temperature $T(x_i,u)$ was set to zero.

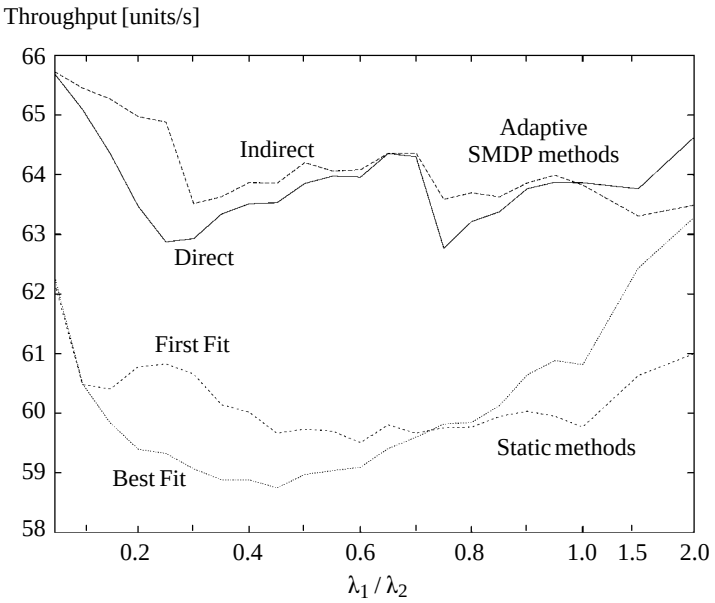

Figure 4: Call level throughput versus arrival rate ratio for different methods.

Some prior knowledge was also needed to complement to the load-sharing policy during adaptation. Experiments with the indirect SMDP scheme showed that one link will always reject narrow-band calls. When $0.25 < \lambda_1/\lambda_2 < 0.75$, this occurred for two links, and when $\lambda_1/\lambda_2 < 0.25$, all narrow-band calls were rejected. The direct scheme did not succeed in finding these complete blocking links, so it had to be predefined in the simulations. A uniform load-sharing policy, set according to the prior knowledge, was used during the initial adaptation period.

The actor/critic parameters were set to $\beta = 0.74$, $\eta_V = 0.1$ and $\eta_\pi = 0.2$. Also, the merit values were initialized to favor *ACCEPT* actions for all $x_i \in X_i$.

The results for the indirect and direct SMDP schemes presented in the diagram in Figure 4 were obtained after 4 adaptation periods, where each adaptation period contained 1 000 and 15 000 simulated call events, for the indirect and direct SMDP method, respectively. The throughput values in the diagram are based on measurements on 300 000 calls events after policy convergence.

The diagram shows that the adaptive SMDP methods yields up to 7% higher long-term revenue than the static methods. The diagram also shows that direct SMDP scheme yields a performance similar to the indirect scheme's.

6 Conclusion

This paper has presented an adaptive scheme, based on reinforcement learning, for a sub-function in ATM network routing called link allocation. The scheme adapts the link allocation policy to the offered Poisson call traffic such that the long-term revenue in maximized.

The experimental results show that the proposed scheme outperforms the static methods and yields a long-term revenue similar to the indirect adaptive SMDP method [1]. The results also show that the adaptation rate of the reinforcement scheme is comparable to the indirect method's.

In our future work, we will consider link allocation of non-Poisson traffic, exploiting the advantages of neural networks as function approximators.

Acknowledgements

The authors would like to thank Mats Gustafsson, Olle Gällmo and Lars Asplund for stimulating discussions. This work was financially supported by ELLEMTEL Telecommunication Systems Laboratories and by NUTEK, the Swedish National Board for Industrial and Technical Development.

References

[1] Z. Dziong and L. Mason,"An Analysis of Near Optimal Call Admission Control and Routing Model for Multi-service Loss Networks", *INFOCOM'92*, Session 2A.1.1, Florence, Italy, May 1992.

[2] R.S. Sutton, "Learning to Predict by the Methods of Temporal Difference", *Machine Learning*, vol. 3, Kluwer Academic Publishers, 1988, pp 9–44.

[3] S.J. Bradtke and M. O. Duff, "Reinforcement Learning Methods for Continuos-Time Markov Decision Problems, in *Advances in Neural Information Processing Systems 8*, D.S. Touretzky, ed., MIT Press, 1995.

[4] A. Barto, R. Sutton and C. Watkins, "Learning and Sequential Decision Making", Report COINS 89–95, Dept. of Computer and Information Science, University of Massachusetts, Amherst, USA, September 1989.