ICICS-PCM 2003
15-18 December 2003
Singapore

Reinforcement-Learning-Based Call Admission Control and Bandwidth Adaptation in Mobile Multimedia Networks

Fei Yu, Vincent W.S. Wong and Victor C. M. Leung

Department of Electrical and Computer Engineering, The University of British Columbia
2356 Main Mall, Vancouver, BC, Canada V6T 1Z4
E-Mail: {feiy, vincentw, vleung}@ece.ubc.ca

Abstract

The availability of bandwidth resources fluctuates much more severely in mobile communication networks compared to wired networks. There is growing interest in developing adaptive multimedia services in mobile communication networks, where it is possible to increase or decrease the bandwidth of individual ongoing flows. This paper studies the issues of call admission control and bandwidth adaptation in such systems. We present a novel approach that models the system as a Markov decision process, and uses a form of reinforcement learning to solve the call admission control and bandwidth adaptation problems without knowledge of the state transition probabilities. More realistic assumptions can therefore be applied to the underlying system model for this approach than in previous schemes. Simulation results demonstrate the effectiveness of the proposed scheme in adaptive multimedia mobile communication networks.

1. Introduction

In recent years, the scarcity and large fluctuations of link bandwidth in wireless networks have motivated the development of adaptive multimedia services where the bandwidth of a connection can be dynamically adjusted to adapt to the highly variable communication environment. Adaptive multimedia services, such as the International Organization for Standardization's (ISO's) Motion Picture Experts Group (MPEG)-4 [1] and the International Telecommunication Union's (ITU's) H.263 [2], are expected to be used extensively in future mobile communication networks, e.g., cellular wireless networks supporting the third generation (3G) universal mobile telecommunications service (UMTS), which can provide flexible radio resource management functions. The bandwidth of a call in these networks can be configured dynamically during the call session [3].

Since radio bandwidth is one of the most precious resources in wireless systems, a bandwidth adaptation (BA) algorithm is required in conjunction with the call admission control (CAC) algorithm in a framework for supporting adaptive multimedia services. BA reallocates the bandwidth of ongoing calls, whereas CAC decides whether to admit or reject new and handoff calls. There are some schemes in the literature addressing CAC and BA for adaptive multimedia services [4-8]. Authors in [4] study the tradeoffs between network overload and fairness in bandwidth adaptation. However, the proposed scheme in [4] does not consider maximizing wireless network utilization and may result in sub-optimal solutions. A near optimal scheme is proposed in [5]. Zaruba *et al.* [6] use a simulated annealing algorithm to find the optimal call-mix selection to maximize the total network revenue under the assumption that future arrivals and departures are known, which may not be realistic in practice. Only one class of adaptive traffic is studied in [7] and [8], and the extension of these schemes to the case of multiple classes may not be an easy task.

In this paper, we formulate the CAC and BA for adaptive multimedia as a Markov decision process (MDP) [9] to find the optimal algorithms that maximize network revenue. The rapid growth in the number of states and the difficulty in estimating the state transition probabilities in practical systems make it very difficult, if not impossible, to use classical methods, such as value iteration or policy iteration [9], to compute the optimal policy. This motivates us to pursue alternative solutions to solve this problem. This paper proposes a scheme using a form of real-time reinforcement learning known as *Q*-learning [10] to solve the MDP. The proposed scheme does not require a priori knowledge of the state transition probabilities associated with the mobile communication network. Therefore, the assumptions behind the underlying system model can be made more realistic than those in previous schemes. Moreover, the proposed scheme can use stochastic approximation to eliminate the need to compute state transition probabilities and to execute complex optimization algorithms. In addition, although quality of service (QoS) constraints, such as handoff dropping probability, are not considered in this paper, it is not difficult to formulate them in the proposed scheme. We will present such a study in a subsequent paper [11].

The rest of this paper is organized as follows. Section 2 describes the reinforcement learning algorithm. Section 3 gives the CAC and BA formulation as well as our new approach to solve this problem. Section 4 presents and discusses the simulation results to demonstrate the effectiveness of our approach. Finally, we conclude this study in Section 5.

This work was supported by the Natural Sciences and Engineering Research Council of Canada under grants RGPIN 261604-03 and RGPIN 44286-00.

2. Reinforcement Learning Algorithm

Reinforcement learning combines concepts from dynamic programming, stochastic approximation via simulation and function approximation. *Q*-learning [10] is one of the most popular reinforcement learning algorithms. We use this algorithm to solve the CAC and BA problems in this paper.

Assume that the environment is a finite-state discrete-time stochastic dynamic system. Let S be the set of possible states, $S = \{s_1, s_2, \ldots, s_n\}$ and A be a set of possible actions, $A = \{a_1, a_2, \ldots, a_m\}$. Based on the state $s_t \in S$, the agent interacting with the environment chooses an action $a_t \in A$ to perform. Then the environment makes a transition to the new state $s_{t+1} = s' \in S$ according to probability $P_{ss'}(a)$ and gives a reward r_t to the agent. The process is repeated.

The goal of the agent is to find an optimal policy $\pi^*(s) \in A$ for each s, which maximizes some cumulative measure of the rewards received over time. The total expected discounted reward over an infinite time horizon is:

$$V^{\pi}(s) = E\left\{\sum_{t=0}^{\infty} \gamma^t r(s_t, \pi(s_t)) \mid s_0 = s\right\}, \qquad (1)$$

where $0 \le \gamma < 1$ is a discount factor and E denotes the expectation. Equation (1) can be rewritten as:

$$V^{\pi}(s) = R(s, \pi(s)) + \gamma \sum_{s' \in S} P_{ss'}(\pi(s)) V^{\pi}(s'), \qquad (2)$$

where $R(s, \pi(s)) = E\{r(s, \pi(s))\}$ is the mean value of reward $r(s, \pi(s))$. The optimal policy π^* satisfies the optimality criterion:

$$V^*(s) = V^{\pi^*}(s) = \max_{a \in A}\left(R(s,a) + \gamma \sum_{s' \in S} P_{ss'}(a) V^*(s')\right). \qquad (3)$$

However, it is difficult to get $R(s,a)$ and $P_{ss'}(a)$ in many practical situations such as the QoS provisioning problem in this paper. *Q*-learning is one of the most popular and effective algorithms for learning from delayed reinforcement to determine the optimal policy. For a policy π, define a Q value as:

$$Q^{\pi}(s,a) = R(s,a) + \gamma \sum_{s' \in S} P_{ss'}(a) V^{\pi}(s'), \qquad (4)$$

which is the expected discounted reward for executing action a at state s and then following policy π thereafter. Let

$$Q^*(s,a) = Q^{\pi^*}(s,a) = R(s,a) + \gamma \sum_{s' \in S} P_{ss'}(a) V^{\pi^*}(s'). \qquad (5)$$

So, $V^*(s) = \max_{a \in A} Q^*(s,a)$. $Q^*(s,a)$ can therefore be written recursively as:

$$Q^*(s,a) = R(s,a) + \gamma \sum_{s' \in S} P_{ss'}(a) \max_{a' \in A}\left(Q^*(s', a')\right). \qquad (6)$$

Then, we have $\pi^*(s) = \arg\max_{a \in A}\left(Q^*(s,a)\right)$ as an optimal policy. The Q-learning process tries to find $Q^*(s,a)$ in a recursive manner using available information. The Q-learning rule is

$$Q_{t+1}(s,a) = \begin{cases} Q_t(s,a) + \alpha \Delta Q_t(s,a) & \text{if } s = s_t \text{ and } a = a_t \\ Q_t(s,a) & \text{otherwise} \end{cases}, \qquad (7)$$

where $\Delta Q_t(s,a) = r_t + \gamma \max_{a' \in A} Q_t(s', a') - Q_t(s,a)$ and α is the learning rate.

3. Reinforcement-Learning-Based CAC and BA for Adaptive Multimedia

3.1 Adaptive Multimedia and Adaptive Mobile Networks

In our adaptive multimedia framework, a multimedia call can dynamically change its bandwidth to adapt to the fluctuating communication environment throughout its call duration. Assume that there are K classes of services in the network. A class i call uses bandwidth among $\{b_{i1}, b_{i2}, \ldots, b_{ij}, \ldots, b_{iN_i}\}$ where $b_{ij} < b_{i(j+1)}$ for $i = 1, 2, \ldots, K$, $j = 1, 2, \ldots, N_i$ and N_i is the highest bandwidth level that can be used by class i call.

The fluctuations in resource availability in mobile communication systems are much more severe compared to those in wired networks. Therefore, the ability of adapting to the communication environment is very important in future mobile communication systems. For example, in UMTS system, a radio bearer established for a call can be dynamically reconfigured during the call session [3]. By reconfiguring the radio bearer, the bandwidth of a call can be changed dynamically during a call session.

3.2 CAC and BA in Adaptive Multimedia Framework

In the adaptive multimedia framework, a bandwidth adaptation (BA) algorithm is required in conjunction with the call admission control (CAC) algorithm. When a cell is in an under-loaded condition, CAC tries to accept every call and BA tries to allocate as much bandwidth as possible to each call. However, network congestion may occur. In this case, calls should be rejected by CAC or degraded to a lower bandwidth by BA. On the other hand, if a call releases its allocated bandwidth due to either call completion or handoff to another cell, some of the calls left in that cell may increase their bandwidth. To decide which call to accept and which call(s) to change the bandwidth are the roles of CAC and BA in the adaptive multimedia framework.

Since forced call terminations due to handoff dropping are generally more objectionable than new call blocking, handoff calls should be given higher priority than new calls. Therefore, we do not admit a new call in an overload cell. In other words, no bandwidth adaptation is used to admit a new call.

We formulate the CAC and BA problems as a Markov decision process (MDP) [9]. However, traditional solutions to MDP, such as value iteration and policy iteration, suffer from two "curses": the "Curse of Dimensionality" and the "Curse of Modeling". The curse of dimensionality occurs in that the algorithms require computation time that is polynomial in the number of states. CAC and BA in mobile multimedia networks involve very large state spaces that make traditional solutions infeasible. The curse of modeling occurs in that in order to apply traditional methods, it is first necessary to express state transition probabilities explicitly; however, they are very difficult to estimate in real networks due to the irregular network topology, different propagation environment, and random user mobility. Therefore, we choose to solve the problem using a reinforcement learning method known as *Q*-learning [10]. This method does not require the explicit expression of the state transition probabilities and can handle MDP problems with large state spaces efficiently. The formulation of this method in solving CAC and BA for adaptive multimedia is presented in following subsection.

3.3 Q-Learning Formulation to Solve QoS Provisioning Problems

In solving CAC and BA problems, the mobile communication system can be considered as a discrete-time event system. These events are modeled as stochastic variables with appropriate probability distributions. We assume that call arrivals including new call arrival events and handoff events follow Poisson distributions. Call holding time is assumed to be exponentially distributed. In order to utilize the *Q*-learning algorithm, we need to identify the system states, actions, and rewards.

States: An event e can occur in a cell c, where e is either a new call arrival, a handoff call arrival, a call termination, or a call handoff to a neighboring cell. At this time, cell c is in a particular configuration $\boldsymbol{x}$ defined by the number of each type of ongoing calls in the cell. $\boldsymbol{x} = (x_{11}, x_{12}, \ldots, x_{ij}, \ldots, x_{KN_K})$, where x_{ij} denotes the number of ongoing calls of class i using bandwidth b_{ij} in cell c for $1 \le i \le K$ and $1 \le j \le N_i$. Recall that K is the number of service classes in the system and N_i is the highest bandwidth level of class i. The configuration and event together determine the state of cell c, $s = (\boldsymbol{x}, e)$.

We assume that each cell has a fixed channel capacity C. The state space is defined as:

$$S = \left\{ s = (\boldsymbol{x}, e) : \sum_{i=1}^{K} \sum_{j=1}^{N_i} x_{ij} b_{ij} \le C \right\}.$$

Actions: When an event occurs, the agent must choose an action according to the state. An action can be denoted as: $a = (a_a, \boldsymbol{a_d}, \boldsymbol{a_u})$, where a_a stands for the admission decision, i.e., admit (a_a = 1), reject (a_a = 0) or no action due to call departure (a_a = -1), $\boldsymbol{a_d}$ stands for the action of bandwidth degradation when a call is accepted and $\boldsymbol{a_u}$ stands for the action of bandwidth upgrade when there is a departure (call termination or handoff to a neighboring cell) from cell c. $\boldsymbol{a_d}$ has the form

$$\boldsymbol{a_d} = \left\{ \left(d_{12}^{1}, \ldots, d_{ij}^{n}, \ldots, d_{KN_k}^{N_k - 1} \right), 1 \le i \le K, 1 < j \le N_i, 1 \le n < j \right\},$$

where d_{ij}^{n} denotes the number of ongoing class i calls using bandwidth b_{ij} that are degraded to bandwidth b_{in}. $\boldsymbol{a_u}$ has the form

$$\boldsymbol{a_u} = \left\{ \left(u_{11}^{2}, \ldots, u_{ij}^{n}, \ldots, u_{KN_k - 1}^{N_k} \right), 1 \le i \le K, 1 \le j < N_i, j < n \le N_i \right\},$$

where u_{ij}^{n} denotes the number of ongoing class i calls using bandwidth b_{ij} that are upgraded to bandwidth b_{in}.

After the action of degradation, the configuration (x_{11}, x_{12}, $\ldots, x_{ij}$, $\ldots$, x_{KN_K}) becomes

$$(x_{11} + \sum_{m=2}^{N_1} d_{1m}^{1}, x_{12} + \sum_{m=3}^{N_1} d_{1m}^{2} - d_{12}^{1}, \ldots, x_{ij} + \sum_{m=j+1}^{N_i} d_{im}^{j} - \sum_{m=1}^{j-1} d_{ij}^{m}, \ldots, x_{KN_K} - \sum_{m=1}^{N_K - 1} d_{KN_K}^{m}).$$

Similarly, after the action of upgrade, the configuration (x_{11}, x_{12}, $\ldots, x_{ij}$, $\ldots$, x_{KN_K}) becomes

$$(x_{11} - \sum_{m=2}^{N_1} u_{11}^{m}, x_{12} + u_{11}^{2} - \sum_{m=3}^{N_1} u_{12}^{m}, \ldots, x_{ij} + \sum_{m=1}^{j-1} u_{im}^{j} - \sum_{m=j+1}^{N_i} u_{ij}^{m}, \ldots, x_{KN_K} + \sum_{m=1}^{N_K - 1} u_{Km}^{N_K}).$$

Rewards: Let r_{ij} be the reward rate of a class i call using bandwidth b_{ij}. The reward rate, $r(s,a)$, can be calculated as:

$$r(s,a)=\sum_{i=1}^{K}\sum_{j=1}^{N_i}x_{ij}r_{ij}\ .$$

Trading off action space complexity with state space complexity: We can see that the action space in our formulation is quite large. It will be time-consuming to find the suitable action given a specific state using RL. We propose a method to trade off action space complexity with state space complexity in the QoS provisioning scheme using a scheme described in [11]. The advantages of doing this are that the action space will be reduced and the extra state space complexity may still be dealt with by using the function approximation.

Suppose that a call arrival event occurs in a cell with state *s*, the action that can be chosen from is ***a*** = (a_a, $d_{12}^1,\ldots,d_{ij}^n,\ldots,d_{KN_k}^{N_k-1}$), where there are at most $W=1+\sum_{i=1}^{K}\sum_{j=2}^{N_i}(j-1)$ components. We can break down the action ***a*** into a sequence of *W* controls a_a, $d_{12}^1,\ldots,d_{ij}^n,\ldots,d_{KN_k}^{N_k-1}$, and introduce some artificial intermediate "states" ($\bar{s}$, a_a), ($\bar{s}$, a_a, d_{12}^1), …, ($\bar{s}$, a_a, $d_{12}^1,\ldots,d_{ij}^n,\ldots,d_{KN_k}^{N_k-1}$), and the corresponding transitions to model the effect of these actions. In this way, the action space is simplified at the expense of introducing $W-1$ additional layers of states and $W-1$ additional *Q* values $Q(\bar{s}$, $a_a)$, $Q(\bar{s}$, a_a, $d_{12}^1)$, …, $Q(\bar{s}$, a_a, $d_{12}^1,\ldots,d_{ij}^n,\ldots,d_{KN_k}^{N_k-2})$ in addition to $Q(\bar{s}$, a_a, $d_{12}^1,\ldots,d_{ij}^n,\ldots,d_{KN_k}^{N_k-1})$. Actually, we view the problem as a deterministic dynamic programming problem with *W* stages. For $w=1,\ldots,W$, we can have a *w*-solution (a partial solution involving just *w* components) for the *w*-th stage of the problem. The terminal state corresponds to the *W*-solution (a complete solution with *W* components). Moreover, instead of selecting the controls in a fixed order, it is possible to leave order subject to choice.

Implementation considerations: In practice, an important issue is how to store the *Q* values in the *Q*-learning algorithm. There are several approaches to representing the *Q* values, among which the lookup table is the most straightforward method. A lookup table representation means that a separate variable *Q*(*s, a*) is kept in memory for each state-action pair (*s, a*). Obviously, when the number of state-action pairs becomes large, the lookup table representation will be infeasible, and some compact representation method is necessary. In this paper, we use the state aggregation approximation method [12]. In this method, the state space *S* is partitioned into *G* disjoint sub-states S_0, S_1, …, S_G. The *Q* value function for all state $s\in S_g$ under action *a* is a constant $\phi(g,a)$ such that

$$\widetilde{Q}(s,a,\phi)=\phi(g,a),\quad \text{if } s\in S_g\ .$$

Then a lookup table can be used for the aggregated problem.

To guarantee the convergence of the *Q*-learning algorithm, each action should be executed in each state an infinite number of times. Therefore, with a small probability p_i, upon the *i*-th decision-making epoch, a decision other than the highest *Q* value is taken. This is called *exploration* [12].

The process of the *Q*-learning-based CAC and BA is shown in Fig. 1. First of all, when an event (either a call arrival or a call departure) occurs, a state *s* can be identified by getting the status of the local cell. Then, CAC and BA can find a set of actions $\{a\}$. Second, look up the aggregated *Q* value table and find a set of *Q* values corresponding to the state *s* and action set $\{a\}$. Third, choose one action from set $\{a\}$ with the maximum *Q* value with probability 1-p_i. Otherwise, perform the exploration. According to the chosen action, the network makes the admission decision and resource adaptation. Fourth, another event occurs and the system reaches another state s'. Finally, the *Q* value is updated according to equation (7).

4. Simulation Results and Discussions

In this section, we present and discuss the simulation results of the proposed scheme. Since we consider homogeneous networks, the performance of the system can be deduced from the performance of a single cell. In the simulations presented in this section, a single cell with bandwidth 2 Mbps is considered. Two classes of flows are considered. Class 1 traffic has three different bandwidth levels, 128, 192 and 256 kbps. 64, 96 and 128 kbps are the three possible bandwidth levels of class 2 traffic. The reward generated by a call is a linear growing function with the bandwidth assigned to the call. Specifically, $r_{ij}=b_{ij}$. We assume that 30% of the offered traffic is from class 1. Moreover, call holding time and cell residence time are assumed to follow exponential distributions with mean 180 and 150 seconds, respectively. The discount factor γ is chosen to be 0.5, and the learning rate α varies with the state-action over time as follows. Each state-action is associated with a learning rate that is inversely proportional to the frequency of the state-action being visited up to the present time.

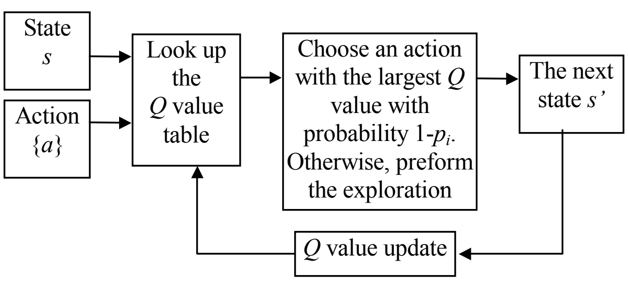
State *s*
Action {*a*}
Look up the *Q* value table
Choose an action with the largest *Q* value with probability 1-*p_i*. Otherwise, preform the exploration
The next state *s'*
Q value update

Fig. 1. The process of *Q*-learning-based CAC and BA

Two other schemes are used for comparisons, the guard channel (GC) scheme [13] and TBA98 [4]. In the GC scheme, a set of bandwidth is reserved permanently for handoff calls. In our simulations, 256 Kbps is reserved for handoff calls. In TBA98, the average bandwidth of currently active flows is used to determine the calls which bandwidth should be increased or decreased in the BA operation.

Fig. 2 shows the rewards of different schemes. We can see that the reinforcement-learning-based scheme yields more reward than the TBA98 or GC schemes. The traditional GC scheme does not use bandwidth adaptation and a call will be rejected if no free bandwidth is available. TBA98 has bandwidth adaptation function and therefore can gain more reward than GC. However, TBA98 does not consider the problem of maximizing the reward. That is why it receives less reward than the proposed scheme. Fig. 3 plots the handoff dropping probability vs. new call arrival rate. We can see that both the reinforcement-learning-based scheme and TBA98 have smaller handoff dropping probability than GC due to the bandwidth adaptation capability in TBA98 and the proposed scheme. However, the handoff dropping probability of all these schemes cannot be kept below some target value when the offered traffic load is high, which is undesirable from users' point of view. We will present a subsequent study in which handoff dropping probability and average allocated bandwidth are formulated as QoS constraints in paper [11].

5. Conclusion

The scarcity and large fluctuations of link bandwidth in wireless networks have motivated the development of adaptive multimedia in mobile communication systems. In this paper, we have formulated the call admission control and bandwidth adaptation in mobile multimedia networks as a Markov decision problem. We have proposed an effective method using reinforcement learning to solve this problem. More realistic assumptions can be applied to the underlying system model for the proposed approach than in previous schemes. The performance of the proposed scheme has been demonstrated by simulations.

References

[1] ISO/IEC 144962-2, "Information technology coding of audio-visual objects: visual," Committee draft, Oct. 1997.

[2] ITU-T H. 263, "Video coding for low bitrate communication," Jan. 1998.

[3] 3GPP, "RRC protocol specification," 3G TS25.331 version 3.12.0, Sept. 2002.

[4] A. K. Talukdar, B. R. Badrinath, and A. Acharya, "Rate adaptation schemes in networks with mobile hosts," in *Proc. ACM/IEEE MobiCom'98*, Oct. 1998.

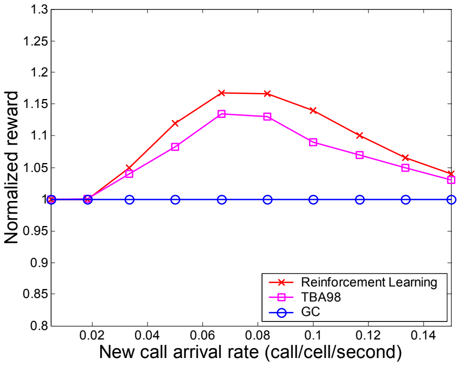

Fig. 2. Normalized reward vs. new call arrival rate

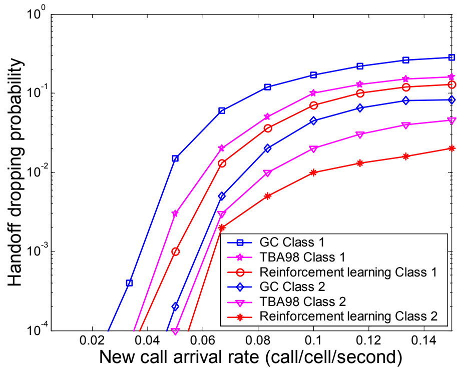

Fig. 3. Handoff dropping probability vs. new call arrival rate

[5] T. Kwon, J. Choi, Y. Choi, and S. K. Das, "Near optimal bandwidth adaptation algorithm for adaptive multimedia services in wireless/mobile networks," in *Proc. IEEE VTC'99-Fall*, vol. 2, Sept. 1999, pp. 874-878.

[6] G. V. Zaruba, I. Chlamtac and S. K. Das, "A prioritized real-time wireless call degradation framework for optimal call mix selection," *Mobile Networks and Applications,* vol. 7, pp. 143-151, 2002.

[7] C. Chou and K. G. Shin, "Analysis of combined adaptive bandwidth allocation and admission control in wireless networks," in *Proc. IEEE Infocom'02*, June 2002.

[8] T. Kwon, Y. Choi, C. Bisdikian and M. Naghshineh, "QoS provisioning in wireless/mobile multimedia networks using an adaptive framework," *Wireless Networks,* vol. 9, pp. 51-59, 2003.

[9] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* New York: Wiley, 1994.

[10] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279-292, 1992.

[11] F. Yu, V.W.S. Wong and V.C.M. Leung, "Efficient QoS provisioning for adaptive multimedia in mobile communication network by reinforcement learning," to appear in *Proc. ACM/SPIE MMCN'04*, Jan. 2004.

[12] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, 1996.

[13] D. Hong and S. S. Rappaport, "Traffic model and performance analysis for cellular mobile radio telephone systems with prioritised and non-prioritised handoff procedures," *IEEE Trans. Veh. Technol.*, vol. VT-35, pp. 77-92, Aug. 1986.