Transactions Papers

A Learning Approach for Prioritized Handoff Channel Allocation in Mobile Multimedia Networks

El-Sayed El-Alfy, *Member, IEEE,* Yu-Dong Yao, *Senior Member, IEEE,* and Harry Heffes, *Fellow, IEEE*

***Abstract*— An efficient channel allocation policy that prioritizes handoffs is an indispensable ingredient in future cellular networks in order to support multimedia traffic while ensuring quality of service requirements (QoS). In this paper we study the application of a reinforcement-learning algorithm to develop an alternative channel allocation scheme in mobile cellular networks that supports multiple heterogeneous traffic classes. The proposed scheme prioritizes handoff call requests over new calls and provides differentiated services for different traffic classes with diverse characteristics and quality of service requirements. Furthermore, it is asymptotically optimal, computationally inexpensive, model-free, and can adapt to changing traffic conditions. Simulations are provided to compare the effectiveness of the proposed algorithm with other known resource-sharing policies such as complete sharing and reservation policies.**

***Index Terms*— Channel allocation, cellular multimedia networks, handoffs, Markov decision processes, dynamic programming, reinforcement learning.**

I. INTRODUCTION

ADVANCES in wireless mobile networks have fueled the rapid growth in research and development to support broadband multimedia traffic and user mobility. Since the available radio resources are scarce, an efficient allocation policy is an essential ingredient in order to accommodate the increasing demands for wireless access. Other motivations include the increasing need to support broadband multimedia traffic and user mobility as well as to ensure quality of service (QoS) requirements in a complex dynamic environment. Issues that make this problem more complicated are the traffic heterogeneity, i.e., diverse traffic characteristics and QoS requirements; as well as the time-varying nature of the network traffic. To increase the allocation efficiency of radio resources, at the network planning level, there is a migration from macro-cells to micro-cells and even pico-cells. As a result a substantial portion of traffic in each cell results from calls migrating from one cell to another (handoffs). However, due to the scarcity of the available resources and the co-channel interference, there is a possibility that a call in progress may be dropped due to handoff failure because of network congestion.

Two important connection-level performance metrics are new call blocking probability and call dropping probability. We consider call dropping as a result of handoff failure due to insufficient resources after initiating a handoff request. These two metrics are always conflicting. From a user point of view, dropping a call in progress due to a handoff failure is more undesirable than rejecting a new call. Therefore, an optimal allocation policy that prioritizes handoff requests over new calls is required to maximize the network throughput and guarantee the quality of service for both active and new calls/sessions. In this paper the QoS is considered to be a function of blocked calls and handoff failures. Multimedia traffic is defined in terms of traffic characteristics and resource requirements, e.g. bandwidth requirements. Resource allocation has been widely studied in several disciplines including engineering, computer science, and operation research. In cellular networks, the efficient use of radio channels is affected by several factors including power control, user mobility pattern, handoff initiation process, and handoff scheduling policy. Two related schemes are the adopted channel assignment scheme among cells and the channel allocation strategy for different traffic classes, i.e., admission control policy.

Several schemes have been proposed in the literature for channel assignment ranging from complete partitioning, e.g., fixed channel assignment (FCA) to complete sharing, e.g., dynamic channel assignment (DCA) and their extensions [1], [2] and references therein. Recently reinforcement learning [3], [4] has been applied to this problem [5], [6]. In these schemes, a call is admitted, with no differentiation between new calls and handoffs, if a channel is available such that the channel reuse constraint is not violated. Since rejecting a new call is less serious than dropping an ongoing call, a tremendous research effort has been devoted to allocation policies that prioritize channel assignment to handoff requests without significantly increasing the new call blocking rates [2], [7]–[14].

One well-known method that prioritizes handoff over new

Manuscript received February 1, 2001; accepted June 22, 2002. The associate editor coordinating the review of this paper and approving it for publication was S. Tekinay.

E.-S. El-Alfy is with the Computer Science Department, King Fahd University of Petroleum and Minerals, Saudi Arabia (email: alfy@kfupm.edu.sa).

Y.-D. Yao and H. Heffes are with the Wireless Information Systems Engineering Laboratory (WISELAB), Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030 USA (email: yyao@stevens.edu; hheffes@stevens.edu).

Digital Object Identifier 10.1109/TWC.2006.01023

call requests is the guard channel, cutoff priority, guard threshold, or trunk reservation policy. The basic idea is to a priori reserve certain number of channels in each cell to handle handoff requests besides allowing the handoff requests to first compete with the new calls for the remaining channels. However since reducing the handoff failures cannot be attained without sacrificing the blocking probability of new calls, the adopted reservation policy has to answer two questions: how many channels should be reserved in each cell and whether they should be a priori reserved [2], [7]–[9] or dynamically changed to adapt to traffic conditions [10], [11]. For a single traffic class, the optimal policy that minimizes a weighted linear function of blocking probabilities has a simple structure of threshold type [12]. The optimal threshold can be determined using analytical models [13] and the theory of Markov decision processes (MDP) [15], [16], e.g., dynamic programming [12], [13], [17] or linear programming [14]. For multiple traffic classes, there is a class of policies known as coordinate convex policies [18]. However, under certain performance indices, the optimal policy is no longer of coordinate convex type [18], [19]. MDP approach can be used to formulate such problem for which the optimal policy search can be performed via dynamic programming or linear programming techniques [20]. All these methods are off-line, static and are based on the assumption that traffic parameters are not changing over time. Furthermore, they rely on the assumption of the knowledge of a perfect analytical state transition and cost model which is not easy to find especially in a complex dynamically changing environment. For such dynamically changing networks supporting several traffic classes with time varying characteristics and requirements, the problem complexity is substantially high, i.e., the state-action space becomes exponentially growing, and exact solutions using traditional optimization techniques become intractable. This is known in literature as the "Curse of Dimensionality" [17]. In this paper we study the application of an average cost reinforcement learning (RL) methodology [21] to develop a channel allocation policy that prioritizes the admission of handoff requests over new call requests. The performance of the proposed algorithm is compared with the optimal guard channel and complete sharing policies. The RL scheme is a stochastic approximation to dynamic programming algorithms (DP) and can be implemented as an on-line "allocation policy" that learns from direct interactions with the network. Also, it has a number of other advantages: simple implementation, model-free, since there is no need for prior knowledge or estimation of the network dynamics, and self-adjusting to time varying traffic conditions. The remainder of this paper is organized as follows. The next section describes the traffic model and the performance measures used; and formally defines the optimization problem to be solved. Section 3 presents the reinforcement learning solution. We first introduce briefly the average cost semi-Markov decision process and then develop an allocation policy based on reinforcement leaning approach for multiple traffic classes and for a single traffic class. Simulations and numerical results are given in Section 4 for two scenarios. In the first scenario we simulate the learning scheme for multiple traffic classes. A comparison with complete sharing policy, both analytical and simulation, are also given. In the second scenario we consider a single traffic class for which we give the analytical exact optimal solution. Then, we run the discrete event simulator for our learning scheme and compare its performance with the corresponding results for complete sharing and optimal guard policies. Finally, in Section 5 we present conclusions and future work.

II. Traffic Model and Problem Description

A. Traffic Model

Consider a cellular network with a fixed number of channels (or bandwidth units) where the service region is divided into small cells with a general spatial layout. Here the concept of channel is used in a generic sense independent of the access technology used whether frequency division multiple access (FDMA), time division multiple access (TDMA), or code division multiple access (CDMA). Multimedia traffic is defined in terms of their traffic characteristics and resource requirements, e.g. bandwidth requirements. We consider K different classes indexed by $k = 1, 2, \ldots, K$ where each class is characterized by a set of traffic parameters: *mean arrival rate*, *bandwidth requirement* (number of channels), *mean service time*, and a *weighting factor* indicating the relative importance of each class or its priority level.

There are two sources of type-k traffic in each cell based on the call originating location: new call arrivals, i.e., calls originated within the cell, and handoff request arrivals, i.e., calls migrating from neighboring cells into that given cell. Under the assumption of spatial uniform traffic conditions and fixed channel assignment strategy between cells, the traffic model of a cellular system can be decomposed into independent cells or clusters [14]. Consider a typical cell with C channels; then the traffic within that cell can be modeled using a C-server system which is an extension of Erlang's loss model [18], [19]. Fig. 1 (a) shows a particular cell within a cellular system and Fig. 1 (b) shows an equivalent heterogeneous channel traffic model in which the C-channels, available in the cell, are represented as C-servers. There are two arrival streams for each class: one representing an aggregate traffic stream for handoff requests into the cell with mean arrival rate λ_{Hk} and the other stream represents the new call arrival within the cell with mean arrival rate λ_{Nk}. The arrival rate matrix is given by,

$$\Lambda = \left[\begin{array}{cccccc} \lambda_1 & \lambda_2 & \cdots & \lambda_k & \cdots & \lambda_{2K} \end{array} \right], \qquad (1)$$

where

$$\lambda_k = \begin{cases} \lambda_{Nk}, & \text{for} \quad 1 \le k \le K \\ \lambda_{H(k-K)}, & \text{for} \quad K+1 \le k \le 2K \end{cases}$$

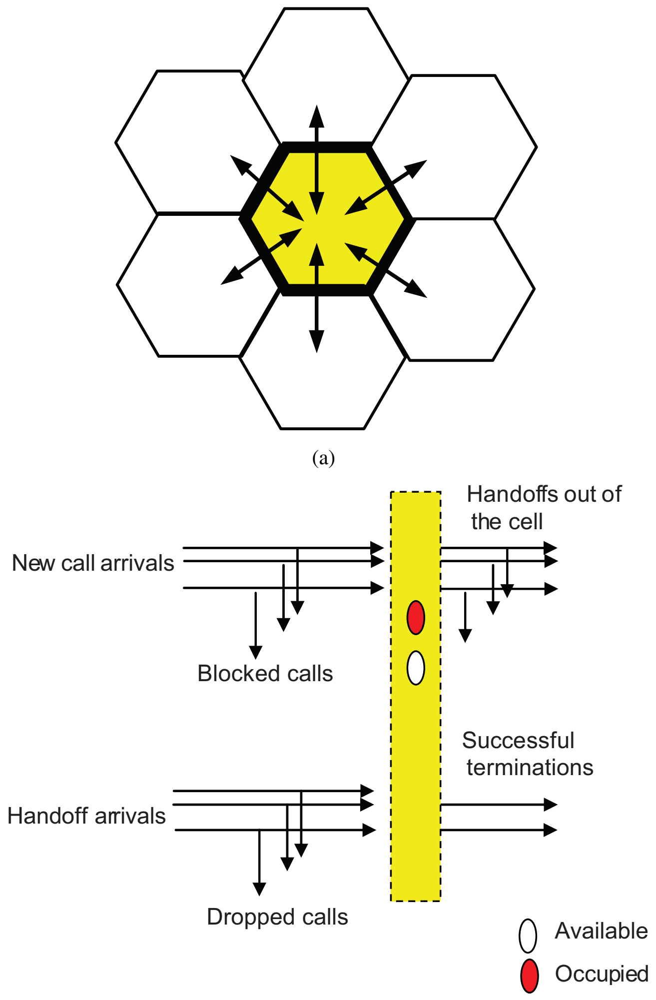
For non-prioritizing techniques such as complete sharing approach, there is no difference between new calls and handoffs. Therefore, the arrival rate vector is given by

$$\Lambda = \left[\begin{array}{cccccc} \lambda_1 & \lambda_2 & \cdots & \lambda_k & \cdots & \lambda_K \end{array} \right], \qquad (2)$$

where

$$\lambda_k = \lambda_{Nk} + \lambda_{Hk}, \forall k \in \{1, 2, \ldots, K\}$$

Each call of type-k, whether a new call or handoff, requests a fixed number of channels given by b_k. These channels



(a)

(b)

Fig. 1. Cellular system layout and traffic model.

are released simultaneously upon the call departure (whether completion or handoff out of the cell). In multimedia networks this may corresponds to a constant bit rate (CBR) service type. However for a variable bit rate (VBR) service class this may represent the peak value or the effective value [22] of the requested bandwidth. The requested band vector is,

$$\mathbf{b} = \begin{bmatrix} b_1 & b_2 & \cdots & b_k & \cdots & b_K \end{bmatrix} \tag{3}$$

Calls leave the cell (the queueing system) as a result of successful call completion or as a result of handoff to one of the neighboring cells. Since the system has finite capacity a new call request or handoff request may be rejected. Without loss of generality of the algorithm, we assume that blocked calls are cleared from the system [18]. We assume that the arrivals of class-k new call and handoff requests are according to mutually independent Poisson processes. We also assume that class-k call duration and the cell dwelling time, i.e., the time until handing-off out of the cell, T_{Dk} and T_{Hk} respectively, are mutually independent and exponentially distributed with means and respectively. T_{Dk} and T_{Hk} are also independent the arrival processes and other traffic classes' times. The channel holding time for class-k, T_k, is the minimum of the call duration and time until handoff and can be expressed as

$$T_k = \min\left(T_{Dk}, T_{Hk}\right) \tag{4}$$

The channel holding time PDF is given by

$$\begin{aligned} F_{T_k}(t) &= \Pr\{T_k \le t\} = 1 - \Pr\{T_{Dk} > t, T_{Hk} > t\} \\ &= 1 - e^{-(\mu_{Dk}+\mu_{Hk})t} \end{aligned} \tag{5}$$

Thus, the channel holding time for class-k traffic is also exponential with mean given by,

$$\frac{1}{\mu_k} = \frac{1}{\mu_{Dk} + \mu_{Hk}} \tag{6}$$

So, the channel holding time vector is expressed as

$$\mathbf{T} = \begin{bmatrix} \mu_1^{-1} & \mu_2^{-1} & \cdots & \mu_k^{-1} & \cdots & \mu_K^{-1} \end{bmatrix} \tag{7}$$

B. Coordinate Convex Policies

The network state, $\mathbf{n}$, can be defined by the vector

$$\mathbf{n} = \begin{bmatrix} n_1 & n_2 & \cdots & n_k & \cdots & n_K \end{bmatrix}, \tag{8}$$

where n_k is the number of active calls of type-k traffic. The system *state space* is a finite set given by

$$S = \left\{ \mathbf{n} \in \{0, 1, \ldots, C\}^K \,|\, \mathbf{n} \cdot \mathbf{b}^T = \sum_{i=1}^{K} n_i b_i \le C \right\} \tag{9}$$

Under a class of policies; known as coordinate convex policies [18], [20], such as complete sharing, complete partitioning and subsets, the admission policy is completely specified with a restricted set of states $\Sigma \subseteq S$. For example, in complete sharing, a request is admitted if the requested number of channels is available regardless of the request type. Therefore, for complete sharing $\Sigma = S$. The natural evolution of the stochastic process $\{\mathbf{n}(t), t \ge 0\}$ represents a *multidimensional* continuous-time Markov chain.

For coordinate convex policies, the admission rule is simple and depends on the next state of the system. Let $\mathbf{n}_k^+$ represent the next state of the network when a new call or handoff of type-k has arrived and has been accepted given that the current state is $\mathbf{n}$, i.e.,

$$\mathbf{n}_k^+ = \begin{bmatrix} n_1 & n_2 & \cdots & n_k + 1 & \cdots & n_K \end{bmatrix} \tag{10}$$

The admission decision depends on what the next state of the system would be if the requested band is granted, and is given by

$$\beta_{Nk}(\mathbf{n}) = \beta_{Nk}(\mathbf{n}) = \begin{cases} 1, & \text{if } \mathbf{n}_k^+ \in \Sigma \\ 0, & \text{elsewhere} \end{cases} \tag{11}$$

where 1 means accept and 0 means reject. This means that the admission rule is the same for both new call and handoff traffic of type-k. To provide preferential treatment for handoffs, the state definition needs to incorporate different variables for new calls and handoffs of each type as if they are two different classes. But this leads to doubling the state space which increases the complexity of any solution procedure. For coordinate convex policies, the transition rate from state $\mathbf{x}$ to state $\mathbf{y}$ is,

$$r_{\mathbf{xy}} = \begin{cases} \lambda_k(\mathbf{x}), & \text{if } \mathbf{x} = \mathbf{n} \in \Sigma \text{ and } \mathbf{y} = \mathbf{n}_k^+ \in \Sigma \\ x_k \mu_k, & \text{if } \mathbf{x} = \mathbf{n} \in \Sigma \text{ and } \mathbf{y} = \mathbf{n}_k^- \in \Sigma \\ 0, & \text{elsewhere} \end{cases} \tag{12}$$

where $\lambda_k(\mathbf{x}) = \lambda_{Nk}\beta_{Nk}(\mathbf{x}) + \lambda_{Hk}\beta_{Hk}(\mathbf{x})$ and $\mathbf{n}_k^- = [\ n_1 \quad n_2 \quad \ldots \quad n_k - 1 \quad \ldots \quad n_K\ ]$.

Following [19], according to the law of conservation of flow, the equilibrium balance equations of any policy can be expressed as:

$$\left[\sum_{k=1}^{K}(\lambda_k(\mathbf{n}) + n_k\mu_k)\right] p(\mathbf{n}) = \sum_{k=1}^{K}(n_k+1)\mu_k p(\mathbf{n}_k^+) + \sum_{k=1}^{K}\lambda_k(\mathbf{n}_k^-)p(\mathbf{n}_k^-), \qquad \sum_{\mathbf{n}\in\ \Sigma} p(\mathbf{n}) = 1 \tag{13}$$

where $p(\mathbf{n})$ is the statistical equilibrium time-average probability that the system occupies state $\mathbf{n}$. This set of equations cannot in general be solved by recurrence to find closed form solutions for the equilibrium state probabilities. However, under coordinate convex policies, the equilibrium state distribution has product form and expressed as [19]:

$$p\left\{\mathbf{n} = (n_1, n_2, \ldots, n_k, \ldots, n_K)\right\} = \prod_{k=1}^{K}\frac{\rho_k^{n_k}}{n_k!}G^{-1}(\Sigma) \tag{14}$$

where,

$$G(\Sigma) = \sum_{\mathbf{n}\in\ \Sigma}\left[\prod_{k=1}^{K}\frac{\rho_k^{n_k}}{n_k!}\right], \quad \rho_k = \frac{\lambda_k}{\mu_k}$$

These results are still applicable even for arbitrary channel holding time distributions due to the insensitivity property [18]. In a most general class of policies, the admission rule may deny service, based on the type of the traffic, even if the requested band is available. In such cases, $\beta_{Nk}(\mathbf{n})$ and $\beta_{Hk}(\mathbf{n})$ may be different and they represent the probability of admitting a new call or a handoff request of type-k respectively when the system is currently in state $\mathbf{n}$. The new call and handoff blocking probabilities experienced by each traffic type can be expressed as,

$$B_{Nk} = 1 - \sum_{\mathbf{n}\in S}\beta_{Nk}(\mathbf{n})p(\mathbf{n})$$
$$B_{Hk} = 1 - \sum_{\mathbf{n}\in S}\beta_{Hk}(\mathbf{n})p(\mathbf{n}) \tag{15}$$

C. The Optimization Problem

The objective of this study is to find a channel allocation policy that minimizes a weighted linear function of new call and handoff blocking probabilities of each type as defined by,

$$P = \sum_{k=1}^{K}\left\{w_{Nk}\frac{\lambda_{Nk}}{\lambda_{Nk}+\lambda_{Hk}}B_{Nk} + w_{Hk}\frac{\lambda_{Hk}}{\lambda_{Nk}+\lambda_{Hk}}B_{Hk}\right\} \tag{16}$$

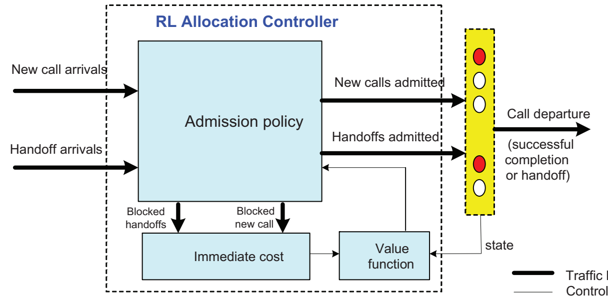


Fig. 2. Traffic model embedded with call admission controller.

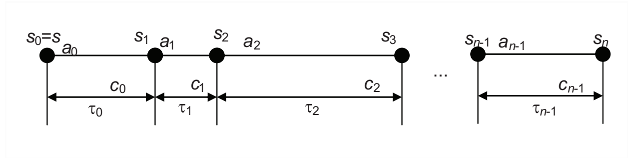


Fig. 3. A sample path given the starting state is s and following a given policy π, where a_t is the action selected at decision epoch t.

where w_{Nk} and w_{Hk} represent the relative weights or the costs incurred when rejecting a new call or handoff of type-k. $w_{Hk} > w_{Nk}$ reflects the fact that rejecting a handoff is more undesirable than blocking a new call. The aim of any admission control strategy is to determine the parameters, $\beta_{Nk}(\mathbf{n})$ and $\beta_{Hk}(\mathbf{n})$, for all states, $\mathbf{n} \in S$. Since the optimal policy may not be coordinate convex, we need to look at a broad class of control policies. However, as pointed in [20], the product form solution does not exist for the equilibrium balance equations outside the class of coordinate convex policies. Another class of algorithms, based on semi-Markov decision problems (SMDP) [15], [16], allows the optimization over all policies. If a perfect system model is known, the optimal solution of SMDP can be attained via dynamic programming (DP) or linear programming (LP) techniques for reasonable problem size. However due to the high dimensionality of the problem, the computational complexity of these traditional approaches is prohibitive. Reinforcement learning, a class of successive approximation methods, can be used to approximately solve the SMDP incrementally on-line without a priori knowledge of the system model. The RL allocation controller learns an admission policy from direct interaction with the network as depicted in Fig. 2. In the next section we present a class of RL channel allocation schemes.

III. Channel Allocation Policy Using Reinforcement Learning

In this section, we briefly review the semi-Markov decision problems (SMDP) under the average cost criterion [15], [16]. Then, the prioritized handoff channel allocation problem among different traffic classes is formulated as an infinite-horizon finite-state SMDP under the average cost criterion. Finally, a class of model-free schemes based on reinforcement learning is used to obtain an asymptotically optimal allocation procedure.

A. Average Cost Semi-Markov Decision Problem

An SMDP is defined by the following three components: a dynamic system, an immediate cost (reward) function, and an objective function. The dynamic system is modeled as a controlled Markov chain. At random points in time (decision epochs) the controller observes the system state and selects an action. As a result the system state will change as a function of the current state, selected action, and external disturbance. Also, the controller receives an evaluative feedback (reinforcement) signal to reflect the immediate (stage) cost incurred. The controller maintains an evaluative (objective) function for each state (state value function), or for each state-action pair (action value function), which is usually an accumulation of the immediate costs incurred over time. There are three most common definitions for the value functions: Finite-horizon expected total cost, infinite-horizon expected discounted total cost, and expected average cost. In the following we use an expected average cost criterion. A deterministic stationary policy is a mapping from states to actions, i.e., $\pi : S \rightarrow A$ where S is a finite set of all states and A is a finite set of all actions. Assume the system starts at state $s_0 = s$ and following policy π, then the long-run average cost, or policy gain, is given by

$$g^{\pi}(s) = \lim_{n \rightarrow \infty} \frac{E\left\{\sum_{t=0}^{n-1} c_t^{\pi}(s)\right\}}{E\left\{\sum_{t=0}^{n-1} \tau_t\right\}} \tag{17}$$

where $c_t^{\pi}(s)$ is the cost incurred at decision time t given that the starting state is s and the implementation policy is π. A typical scenario for the evolution of the embedded Markov chain is illustrated in Fig. 3.

For ergodic Markov decision processes, the average cost exists and is independent of the starting state [16], i.e., $g^{\pi}(\mathbf{x}) = g^{\pi}(\mathbf{y}) = g^{\pi}\ \forall \mathbf{x}, \mathbf{y} \in S$ and $\forall \pi \in \Pi$.

Here Π is the space of feasible policies and g^{π} is the long-term average cost starting from any initial state and following policy π. The controller objective is to determine a policy in order to minimize the long-run average cost. Formally, we are seeking a policy π^* with corresponding long-run average cost which is optimal, i.e., $g^* = \min_{\pi \in \Pi} g^{\pi}$.

According to Bellman's optimality principle [15], for finite state and action spaces, an optimal policy has the property that, whatever the initial state and decision are, the remaining decisions must form an optimal policy with regard to the resulting state from the first transition. The optimality equations for the average cost semi-Markov decision process have the following recursive form:

$$h^*(x) = \min_{a \in A_i} \left\{ c(x,a) - g^* \tau(x,a) + \sum_{y \in S} p(y|x,a) h^*(y) \right\}, \quad \forall x \in S \tag{18}$$

where $h^*(x)$ is an optimal average-adjusted state value function $h^* : S \rightarrow \Re, c(x,a)$ is the expected immediate cost incurred when being in state s and action a is selected; and $\tau(x,a)$ is the average sojourn time until the next decision epoch when being in state *s* and action *a* is selected. The state transition probability $p(y|x,a)$ is defined as the probability that the state at the next decision epoch is y given that the system currently in state x and performing action a. Solving these equations results in the optimal values g^* and $h^*(x) \forall x \in S$. Knowing $h^*(x) \forall x \in S$, the corresponding optimal policy is determined by

$$\pi^*(x) = \arg\min_{a \in A_i} \{ c(x,a) - g^* \tau(x,a) + \sum_{y \in S} p(y|x,a) h^*(y) \}, \ \forall x \in S \tag{19}$$

Another approach is to define a value function for each available action in each state, or Q-values [23], as follows

$$Q^*(x,a) = c(x,a) - g^* \tau(x,a) + \sum_{y \in S} p(y|x,a) \times \min_{b \in A_y} \{ Q^*(y,b) \}, \ \forall x \in S, x \in A_x \tag{20}$$

where $h^*(x) = \min_{a \in A_x} Q^*(x,a), \ \forall a \in S$.

The optimal action in each state is determined according to

$$\pi^*(x) = \arg\min_{a \in A_x} Q^*(x,a) \tag{21}$$

B. Reinforcement Learning Schemes

The computational complexity hinders the classical optimization methods for solving the optimality equations, such as linear programming and dynamic programming, to scale well for practical dynamic networks. Reinforcement learning can be used to learn an optimal policy on-line from direct interactions with the environment by successively approximating dynamic programming algorithms. RL is a class of algorithms that are capable of improving their performance incrementally. There are two most common techniques for solving reinforcement learning problems: Sutton's temporal difference [24] and Watkins' Q-learning [23].

Temporal difference learning: The temporal difference learning scheme is a successive approximation of the asynchronous dynamic programming value iteration algorithm for finding solutions of the system of equations (18). It learns the state value functions iteratively using the sample values, instead of the expected values, by the following incremental averaging rule

$$h_{new}(x) = (1-\alpha) h_{old}(y) + \alpha \{ c(x,a,y) - \hat{g} \tau(x,a,y) + h_{old}(y) \} \tag{22}$$

where $\alpha \in [0,1]$ is the learning rate, or step size parameter and the setting for α may be fixed or diminishing over time.

Q-learning: The controller uses the sample information to incrementally update the state-action value function, to solve the system of equations (20), using temporal difference as follows

$$\hat{Q}_{new}(x,a) = (1-\alpha) \hat{Q}_{old}(x,a) + \alpha \{ c(x,a,y) - g^* \tau(x,a,y) + \min_{b \in A_y} (\hat{Q}_{old}(y,b)) \} \tag{23}$$

where again $\alpha \in [0,1]$ is the learning rate. At each decision epoch, the action selection is performed based on the estimated Q-values in a number of ways to balance the exploration and exploitation tradeoff. One selection rule is known as the greedy selection policy, where no exploration is made, an action is selected in state x such that

$$\pi(x) = \arg\min_{a \in A_x} \{\hat{Q}(x,a)\} \tag{24}$$

Another approach, called ε-greedy, is to select the greedy action with a high probability and with a small probability, ε, uniformly select among other actions. The later policy has an advantage over the greedy policy with respect to the the rate of convergence since it allows the Q-values to be updated for non-greedy actions and maintains a balance between exploiting information and exploring.

Average cost estimation: Since the controller has no information about the long-run average cost g^* it can be similarly estimated online using the following update rule

$$\hat{g}_{new} = (1-\beta)\hat{g}_{old} + \beta \frac{c(x,a,y) + \min_{b \in A_y} \hat{Q}(y,b) + \min_{a \in A_x} \hat{Q}(x,a)}{\tau(x,a,y)} \tag{25}$$

where $\beta \in [0,1]$ is another learning rate. Another way to estimate g^*, using accumulated costs and times, is

$$\hat{g} = \frac{\sum_{i=0}^{n-1} c\left(s_i, a_i, s_{i+1}\right)}{\sum_{i=0}^{n-1} \tau\left(s_i, a_i, s_{i+1}\right)} \tag{26}$$

The above estimates can update the value of g^* at each decision epoch or only when selecting a greedy action.

The asymptotic convergence of reinforcement learning algorithms is based on the assumption that all available actions in each state are tried infinitely often. The selection of the learning rates and the exploration-exploitation affects the rate of convergence. For large state spaces, the Q-values can be represented as a parameterized function using function approximation methods, e.g., neural networks [4].

1) Multiple Traffic Class Case: In the multiple traffic classes with prioritized handoff problem, one obvious approach to formulate the allocation problem is to treat the new call and handoff of type-k traffic, $1 \le k \le K$, as two distinct classes. In what follows we use another more efficient formulation based on the state aggregation. In aggregated states we need only to use one variable for each traffic type to indicate the number calls of that type, both new calls and handoffs, in progress. This change results in a dramatic reduction in the cardinality of the state space. Also, it allows the states to be visited more often and updates the value functions more often. Therefore, more accurate results can be obtained. In this case, the *system state* at time, t, is defined as

$$\mathbf{n}(t) = \begin{bmatrix} n_1(t) & n_2(t) & \cdots & n_k(t) & \cdots & n_K(t) \end{bmatrix}, \tag{27}$$

where $n_k(t)$ is the number of on-going new calls of type-k traffic at time t if $1 \le k \le K$. The state space is a finite set given by,

$$S = \left\{ \mathbf{n} \in \{0,1,\ldots,C\}^K \,|\mathbf{n} \cdot \mathbf{b}^T = \sum_{k=1}^{K} n_k b_k \le C \right\} \tag{28}$$

The *decision epochs* are defined to occur at discrete times corresponding to the occurrence of external events (disturbance). There are three external events for type-k traffic: new call arrival, handoff arrival, and call departure. Let $\mathbf{e} = [e_1 \quad e_2 \quad \ldots \quad e_K \quad \ldots \quad e_{2K}]$ be an event vector where $e_k \in \{1,0,-1\}$ for $1 \le k \le K$ corresponds to type-k new call arrival, $e_k = 1$, type-k call departure, $e_k = -1$, or no change, $e_k = 0$; and $e_k \in \{1,0\}$ for $K+1 \le k \le 2K$ corresponds to type-k handoff arrival, $e_k = 1$, or no change, $e_k = 0$. At each decision epoch the controller observes the system state and, based on the event type, selects an action. Let $A_{\mathbf{x},\mathbf{e}}$ be a finite set of the available actions in state $\mathbf{x}$ when event $\mathbf{e}$ occurs and given by

$$A_{\mathbf{x},\mathbf{e}} = \begin{cases} \{1,0\}, & \text{if } e_k = 1 \text{ and } \mathbf{x}_k^+ \in S \\ \{1\}, & \text{if } e_k = 1 \text{ and } \mathbf{x}_k^+ \notin S \\ \{0\}, & e_k = -1 \end{cases} \tag{29}$$

where $a = 1$ means reject and $a = 0$ means otherwise (accept or no-action).

Given the current state $\mathbf{x}$, the current event $\mathbf{e}$ and the selected action $\mathbf{a}$; then the next state $\mathbf{y}$ is deterministically given by $\mathbf{y} = f(\mathbf{x},\mathbf{e},a)$ where:

$$\mathbf{y} = \begin{cases} \mathbf{x}, & \text{if } e_k = 1 \text{ and } a = 1 \\ \mathbf{x}_k^+, & \text{if } e_k = 1 \text{ and } a = 0 \\ \mathbf{x}_k^-, & e_k = -1 \end{cases} \tag{30}$$

The time from one decision epoch to the next decision epoch is a continuous random variable with a conditional probability distribution, $F_\tau(t|\mathbf{x})$, given the current state $\mathbf{x}$ immediately after the decision. For the above settings, i.e., exponential inter-arrival and service times, $F_\tau(t|\mathbf{x})$ is also exponential with expected time until the next decision epoch (next event) given as

$$\tau(\mathbf{x}) = \left[\sum_{k=1}^{K} x_k \mu_k + \sum_{k=1}^{2K} \lambda_k \right]^{-1} \tag{31}$$

The probability of the next event being an arrival or departure of type-k, given the current state $\mathbf{x}$, is expressed as

$$p(\mathbf{x},\mathbf{e}) = \begin{cases} \lambda_k \tau(\mathbf{x}), & \text{if } e_k = 1 \text{ and } 1 \le k \le 2K \\ x_k \mu_k \tau(\mathbf{x}), & \text{if } e_k = -1 \text{ and } 1 \le k \le K \end{cases} \tag{32}$$

When a request of type-k is rejected, the immediate (stage) cost, $c(\mathbf{x},\mathbf{e},a)$, incurred is

$$\begin{cases} w_{Nk}, & \text{if } ((e_k = 1 \text{ and } a = 1) \text{ and } 1 \le k \le K) \\ w_{H(k-K)}, & \text{if } ((e_k = 1 \text{ and } a = 0) \text{ and } K+1 \le k \le 2K) \\ 0, & \text{elsewhere} \end{cases} \tag{33}$$

A deterministic stationary allocation policy is a mapping from states and events to actions, i.e., $\pi : SxE \to A$ where E is a finite set of all events and A is a finite set of all actions.

The optimality equations for the average cost semi-Markov decision process will have the following recursive form:

$$h^*(\mathbf{x}) + \mathbf{g}^* \tau(\mathbf{x}) = E_e \left\{ \min_{a \in A_i} \{ c(\mathbf{x},\mathbf{e},a) + h^*\left(f\left(\mathbf{x},\mathbf{e},a\right)\right) \} \right\}, \quad \forall \mathbf{x} \in S \tag{34}$$

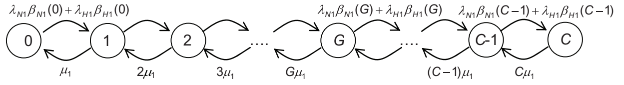


Fig. 4. Generic allocation strategy: transition rate diagram for a single traffic class.

where $h^*(\mathbf{x})$ is an optimal state dependent value function $h^* : S \rightarrow R$. Solving these equations results in the optimal values $h^*(\mathbf{x}), \forall \mathbf{x} \in S$ and g^*. Knowing $h^*(\mathbf{x}), \forall \mathbf{x} \in S$, the corresponding optimal policy is determined by

$$\pi^*(\mathbf{x},\mathbf{e}) = \arg \min_{a \in A_x, e} \{c(\mathbf{x},\mathbf{e},a) + h^*(f(\mathbf{x}, \mathbf{e}, a))\}, \quad \forall \mathbf{x} \in S \text{ and } \mathbf{e} \in E \tag{35}$$

We use a temporal difference approach TD(0) [24] to learn an optimal policy. The relative value functions are estimated on-line from the generated state-action sample sequence. Initially the controller parameters, value functions and policy gain, are set to some values, e.g., zeros. At each decision epoch the system state is observed and the state value function is updated according to

$$h_{new}(\mathbf{x}) = (1-\alpha)h_{old}(\mathbf{x}) + \alpha\{c(\mathbf{x},\mathbf{e},a) - g\tau(\mathbf{x}) + h_{old}(f(\mathbf{x}, \mathbf{e}, a))\}, \tag{36}$$

where again $\alpha \in [0, 1]$ is the learning rate.

A number of variations are possible for the above basic problem formulation approach by using different state definitions, e.g., expanded states, and/or changing the decision epochs to correspond to all events or arrivals only.

State Expansion: We expand the state vector to incorporate the event vector as part of the state definition, i.e., $s = (\mathbf{n}, \mathbf{e})$. This change results in a dramatic reduction in the action space cardinality although it increases cardinality of the state space. When the requested band is available, two actions are possible in each state to admit or reject the call request. This is useful reduction if we learn use Q-learning to learn the action value functions instead of the state value functions.

Decision Epochs: Changing the decision epochs to correspond to the arrivals only, where actual decisions are required, results in reduction in the cardinality of the event space. In this case the system state may change several times between decision epochs due to call departures and therefore the next state is stochastically determined by the state transition probability. This change results in complications in the dynamic programming algorithms since they need a prior knowledge of a perfect transition and cost model. However, it helps the learning schemes since they are model-free and there is no need to learn or update the value functions at the departure events.

2) The Single Traffic Class Case: For a single traffic class, the transition rate diagram is shown in Fig. 4 for a generic allocation policy. The optimal allocation policy that prioritizes handoffs has a simple structure of guard threshold type.

In [12], it is proved that a guard channel approach is optimal for a single traffic class, therefore $\beta_{H1}(i) = 1$, for $0 \leq i \leq C$, and $\beta_{N1}(i) = 1$, for $0 \leq i \leq G$ and zero otherwise. G is an optimal threshold value.

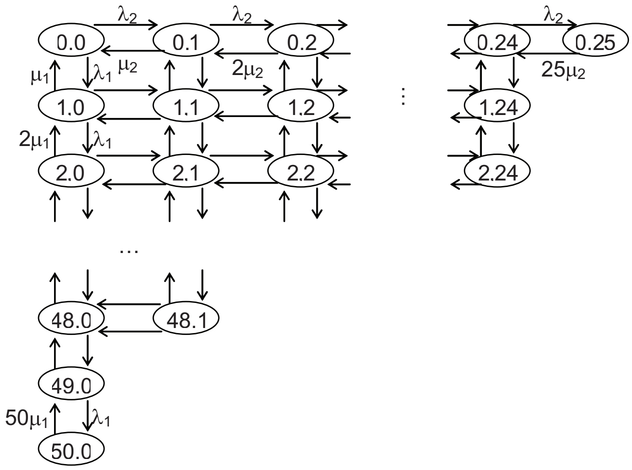


Fig. 5. Complete sharing: Transition rate diagram for two traffic classes with different characteristics.

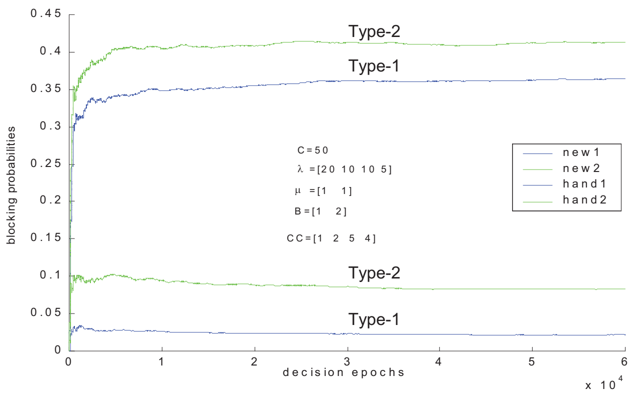


Fig. 6. Simulation results of the learning algorithm for two-class example: new call and handoff blocking probabilities for each traffic type.

RL learning can be used to determine an asymptotically optimal allocation policy on-line. Although we still can use the above proposed scheme for multiple traffic classes when $K = 1$, in this subsection we propose another scheme for which the controller only interacts with network when a new call arrives. Following the results of [12], it is required only to search for a control policy for new call arrivals. Therefore, the allocation controller always accepts handoff calls if the requested band is available. However, it uses reinforcement learning to learn an admission policy for new call arrivals. The system state is defined as the number of calls in progress (aggregated state) which result in reducing the state space cardinality to be $|C|$ where C is the total number of channels available in the cell. The decision epochs correspond to new call arrivals. The system state may change several times between two decision epochs as a result of handoff arrivals or call departures which are considered as external disturbance. The system *state space* is a finite set $S = \{0, 1, 2, \ldots, C\}$. The *action set* available in each state is a finite set $A_s = \{1 = reject, 0 = admit\}$ for $s \in \{0, 1, 2, ..., C-1\}$ and $A_s = \{0 = reject\}$ for $s \in \{C\}$. A deterministic stationary policy is a mapping from states to actions $\pi : S \rightarrow A$. Since the state transitions are now random, we use the average adjusted Q-learning algorithm for continuous time SMDP to learn an optimal admission policy for new call arrivals. When a new call arrives, the system state is observed and a decision is made based the estimated action value functions.

IV. Simulation and Numerical Results

In this section the performance of the learning algorithm will be demonstrated and compared with complete sharing and optimal guard channel reservation. We use a discrete event simulator to generate the traffic streams for new call and handoff call requests according to mutually independent Poisson processes. The channel holding times are exponentially distributed.

A. Multiple Traffic Class Case

Consider a particular cell within a cellular network assigned a fixed set of channels $C = [50]$. There are two types of traffic classes, $K = 2$, each type has two differentiated arrival streams: new calls and handoff requests. New calls and handoffs of type-k arrive according to mutually independent Poisson processes. The arrival rate matrix is

$$\Lambda = \left[\begin{array}{cccc} \lambda_{N1} & \lambda_{N2} & \lambda_{H1} & \lambda_{H2} \end{array}\right] = \left[\begin{array}{cccc} 20 & 10 & 10 & 5 \end{array}\right] \tag{37}$$

Each call of type-k, whether a new call or handoff, requests a number of channels given by b_k and holds these channels for an exponentially distributed time with mean $1/\mu_k$ where

$$T = \left[\begin{array}{cc} \mu_1^{-1} & \mu_2^{-1} \end{array}\right] = \left[\begin{array}{cc} 1 & 1 \end{array}\right] \tag{38}$$

and,

$$b = \left[\begin{array}{cc} b_1 & b_2 \end{array}\right] = \left[\begin{array}{cc} 1 & 2 \end{array}\right] \tag{39}$$

For complete sharing policy, there is no difference between new calls and handoffs. Therefore, the arrival rate vector is given by

$$\Lambda = \left[\begin{array}{cc} \lambda_1 & \lambda_2 \end{array}\right] = \left[\begin{array}{cc} 30 & 15 \end{array}\right] \tag{40}$$

where $\lambda_k = \lambda_{Nk} + \lambda_{Hk}, \ \forall k$. Let $\mathbf{n} = (n_1, n_2)$ be the aggregated state of the system, where n_k is the number of active calls of type-k in the cell. The system state space is a finite set given by

$$S = \left\{ (n_1, n_2) \,|\, \sum_{k=1}^{2} n_k b_k \leq C \right\} \tag{41}$$

The transition rate diagram for a two-dimensional Markov chain is shown in Fig. 5. The complete sharing admission decision rule for type-k traffic is,

$$a = \begin{cases} 0, & \text{if } \mathbf{n}_k^+ \in S \\ 1, & \text{elsewhere} \end{cases} \tag{42}$$

where $a = 0$ means accept and $a = 1$ means reject. The steady-state blocking probabilities are given by

$$P_{B1} = \sum_{\left\{(n_1,n_2)|0\leq n_2\leq \lfloor \frac{C}{b_2} \rfloor, n_1 = \lfloor \frac{C-n_2 b_2}{b_1} \rfloor \right\}} P(n_1, n_2) \tag{43}$$

and

$$P_{B2} = \sum_{\left\{(n_1,n_2)|0\leq n_1\leq \lfloor \frac{C}{b_1} \rfloor, n_2 = \lfloor \frac{C-n_1 b_1}{b_2} \rfloor \right\}} P(n_1, n_2) \tag{44}$$

where is the largest integer smaller than x.

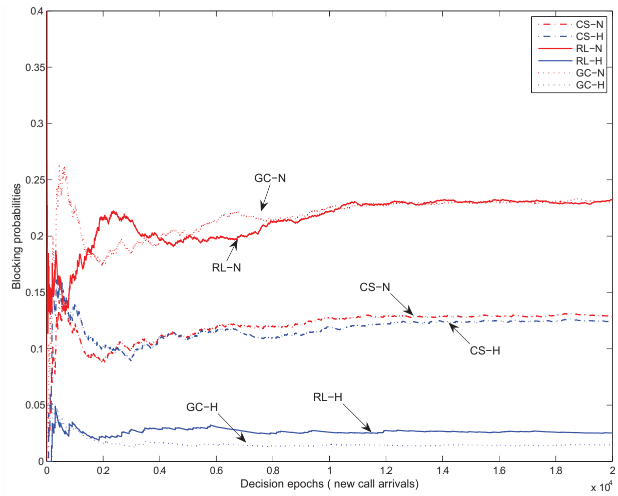


Fig. 7. Simulation results for comparing the blocking probabilities of the three policies for same traffic scenario over a single run.

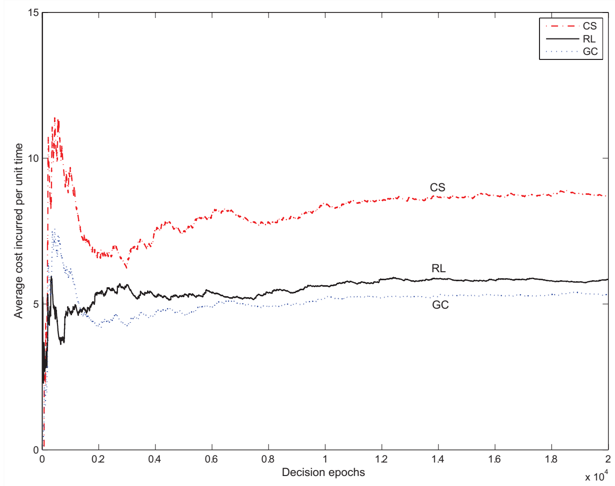


Fig. 8. Long-term average cost incurred per unit time over a single run with same traffic scenario as Fig. 7.

For the given numerical values above, the analytical values for the blocking probabilities are $P_{B1} = 0.1622$ and $P_{B2} = 0.3063$. The allocation policy for this typical example is obtained through the learning algorithm and its simulation performance is depicted in Fig. 6. The controller observes the system state at each event and learns the aggregated state value functions on-line. When a call arrival of type-k occurs, the controller selects an action based on the estimated average-adjusted states values.

B. Single Traffic Class Case

We consider a particular cell within a cellular network with $C = 30$ channels, $\lambda_N = 20$, $\lambda_H = 10$, $\mu_C = 1$, $w_N = 1$ and $w_H = 5$. The analytical solution reveals the optimal threshold $G = 28$ and the corresponding blocking probabilities $P_B^H = 0.0185$ and $P_B^N = 0.2354$.

The simulation-based performance for the three policies (CS: complete sharing, GC: guard channel, and RL: reinforcement learning) is depicted in Figs. 7-12. Fig. 7 shows a single run accumulated blocking probabilities. For optimal

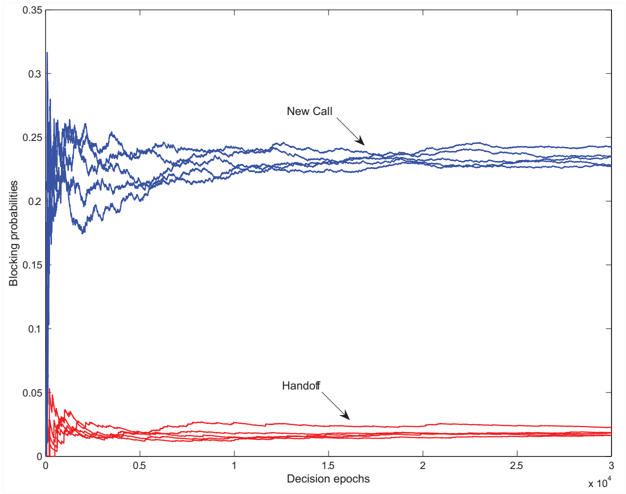


Fig. 9. Simulation results for guard channel policy over 5 runs for different traffic scenarios.

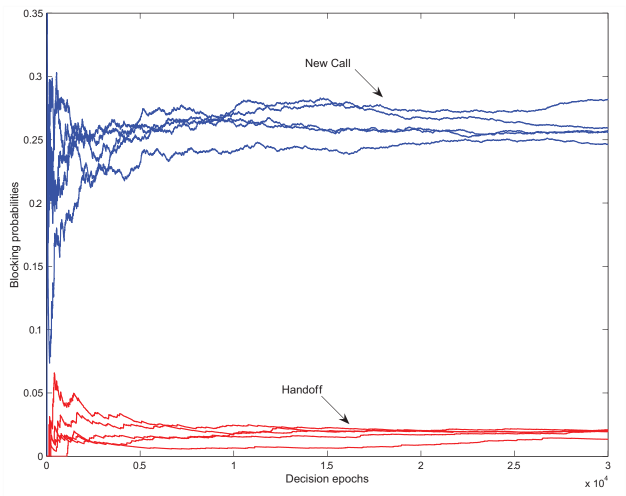


Fig. 10. Simulation results for reinforcement learning policy over 5 runs for different traffic scenarios.

guard threshold, the blocking probabilities are approximately $P_B^H = 0.0147$ and $P_B^N = 0.2305$; and for the learning approach $P_B^H = 0.0248$ and $P_B^N = 0.2325$ which are very close to the analytical solutions. Fig. 8 compares the long-run average cost incurred per time step. Again the performance of the learning approach is very close to the optimal guard threshold. Figs. 9 and 10 depict the blocking probabilities for the guard channel and learning policies for five simulation runs. Figs. 11 and 12 show the corresponding average over the five runs. As seen the learning approach has a comparative performance to the optimal guard threshold.

V. Conclusions and Future Work

The call admission control in cellular mobile networks with prioritized handoffs has been formulated as an average cost continuous-time Markov decision process and a reinforcement learning approach for finding a near-optimal admission policy has been proposed. A key finding of this study is that the reinforcement leaning algorithm has a comparative performance, for a single traffic class case, to the optimal

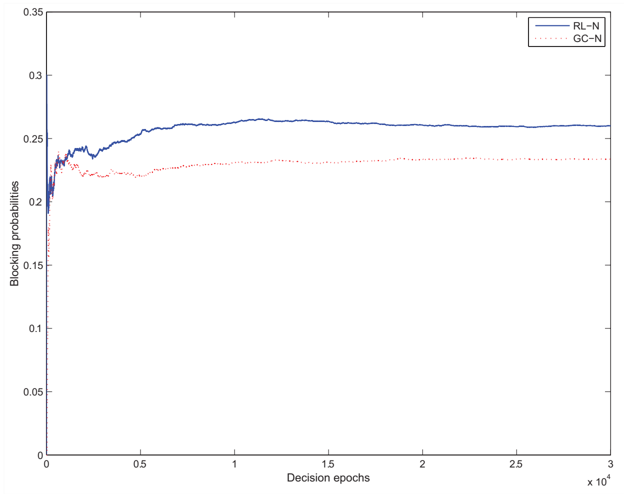


Fig. 11. Simulation results for guard channel policy and learning: new call average blocking probabilities after 5 runs for different scenarios of Figures 9 and 10.

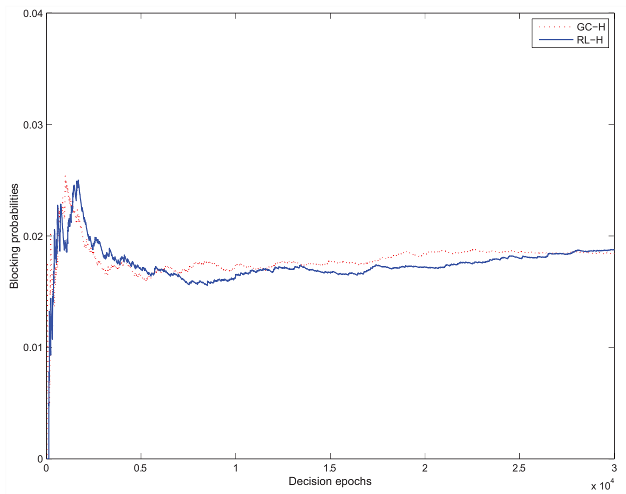


Fig. 12. Simulation results for comparing handoff average dropping probabilities for learning and optimal guard policies – the average is over same 5 runs as Fig. 11.

guard threshold policy. For multiple traffic classes the learning algorithm generalizes the concept of guard threshold to guard states. This paper is part of a major study in which we are currently exploring the application of the learning algorithm for multimedia cellular mobile networks under diverse QoS constraints. Open research topics include the study of the handoff-prioritized channel allocation among multiple traffic classes in multi-cell systems. Others are to incorporate the information gained about the system model over time in the learning process, and to study the performance of the learning algorithm for non-stationary conditions.

Acknowledgment

The first author would like to thank King Fahd University of Petroleum and Minerals (KFUPM) for their support during this work.

References

[1] G. L. Stüber, *Principles of Mobile Communication*. Norwell, MA: Kluwer Academic Publishers, 1996.

[2] S. Tekinay and B. Jabbari, "Handover and channel assignment in mobile cellular networks," *IEEE Commun. Mag.*, vol. 29, no. 4, pp. 42-46, Apr. 1991.
[3] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.
[4] D. P. Bertsekas and J. N. Tsitsiklis, *Neurodynamic Programming*. Belmont, MA: Athena Scientific, 1996.
[5] S. P. Singh and D. P. Bertsekas, "Reinforcement learning for dynamic channel allocation in cellular telephone systems," in *Advances in NIPS*, pp. 974-980 (Cambridge, MA: MIT Press, 1997).
[6] J. Nie and S. Haykin, "A Q-learning based dynamic channel assignment technique for mobile communication systems," *IEEE Trans. Veh. Technol.*, vol. 48, no. 5, pp. 1676-1687, Sep. 1999.
[7] S.-H. Oh and D.-W. Tcha, "Prioritized channel assignment in a cellular radio networks," *IEEE Trans. Commun.*, vol. 40, no. 7, pp. 1259-1269, July 1997.
[8] R. Guerin, "Queueing-blocking system with two arrival streams and guard channels," *IEEE Trans. Commun.*, vol. 36, no. 2, pp. 153-163, Feb. 1988.
[9] D. Hong and S. S. Rappaport, "Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and nonprioritized handoff procedures," *IEEE Trans. Veh. Technol.*, vol. 35, pp. 77-92, July 1986.
[10] S. Boumerdassi and A. Beylot, "Adaptive channel allocation for wireless PCN," *Mobile Networks and Applications*, vol. 4, no. 2, pp. 111-116, June 1999.
[11] O. Yu and V. Leung, "Adaptive resource allocation for prioritized call admission over an ATM-based wireless PCN," *IEEE J. Select. Areas Commun.*, vol. 15, no. 7, pp. 1208-1225, Sep. 1997.
[12] D. T. R. Ramjee and R. Nagarajan, "On optimal call admission control in cellular networks," *Wireless Networks*, vol. 3, no. 1, pp. 29-41, Jan. 1997.
[13] M. Saquib and B. Yates, "Optimal call admission to a mobile cellular network," in *Proc. IEEE Veh. Technol. Conf.*, July 1995, pp. 190-194.
[14] C. J. Ho and C. T. Lea, "Improving call admission policies in wireless networks," *Wireless Networks*, vol. 5, no. 4, pp. 257-265, July 1999.
[15] M. L. Putterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York: Wiley, 1994.
[16] H. C. Tijms, *Stochastic Models: An Algorithmic Approach.* New York: Wiley, 1994.
[17] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA: Athena Scientific, 1995.
[18] J. S. Kaufman, "Blocking in a shared resource environment," *IEEE Trans. Commun.*, vol. 29, no. 10, pp. 1474-1481, Oct. 1981.
[19] R. B. Cooper, *Introduction to Queueing Theory*. New York: Elsevier Science Publishing Co. Inc., 1981.
[20] K. W. Ross and D. H. Tsang, "Optimal circuit access policies in an ISDN environment: A Markov decision approach," *IEEE Trans. Commun.*, vol. 37, no. 9, pp. 934-939, Sep. 1989.
[21] S. Mahadevan, "Average reward reinforcement learning: Foundations, algorithms and empirical results," *Machine Learning*, vol. 22, no. 1, pp. 159-196, Mar. 1996.
[22] R. Guerin, H. Ahmadi, and M. Naghshineh, "Equivalent capacity and its application to bandwidth allocation in high-speed networks," *IEEE J. Select. Areas Commun.*, vol. 9, no. 7, pp. 968-981, Sep. 1991.
[23] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279-292, May 1992.
[24] R. S. Sutton, "Learning to predict by the method of temporal differences," *Machine Learning*, vol. 3, no. 1, pp. 9-44, Aug. 1988.

El-Sayed El-Alfy (S'00-M'01) received his M.S. degree in Computer Science and Ph.D. in Computer Engineering, both from Stevens Institute of Technology, Hoboken, NJ, in 2002. He is currently working as an Assistant Professor in the Computer Science Department, King Fahd University of Petroleum and Minerals, Saudi Arabia. He is teaching courses on networking, network design, client/server programming, and multimedia data compression. He is serving in a number of standing departmental and college committees. From 2002-2004, he was appointed as an Assistant Professor at Tanta University, Egypt; and also as Adjunct Assistant Professor at Delta Academy of Computers, Egypt and a Consultant for ITC, Canadian Project in Egypt. From 2001-2002, he was with Lucent Technologies Inc. as a Member of Technical Staff and was a post-doctoral fellow at Stevens Institute of Technology. From 1998-2001, he was an Affiliated Instructor at Stevens. From 1992-1997, he worked as a Graduate Assistant in the Department of Computer and Control Engineering, Tanta University, Egypt. His current research interests include mobile and distributed computing systems: performance analysis, design, optimization, management and applications; and applications of soft computing in network-related problems.

Yu-Dong Yao (S'88-M'88-SM'94) received the B.Eng. and M.Eng. degrees from Nanjing University of Posts and Telecommunications, Nanjing, China, in 1982 and 1985, respectively, and the Ph.D. degree from Southeast University, Nanjing, China, in 1988, all in electrical engineering.

From 1989 and 1990, he was at Carleton University, Ottawa, Canada, as a Research Associate working on mobile radio communications. From 1990 to 1994, he was with Spar Aerospace Ltd., Montreal, Canada, where he was involved in research on satellite communications. From 1994 to 2000, he was with Qualcomm Inc., San Diego, CA, where he participated in research and development in wireless code-division multiple-access (CDMA) systems. He joined Stevens Institute of Technology, Hoboken, NJ, in 2000, where he is an Associate Professor in the Department of Electrical and Computer Engineering and a Director of the Wireless Information Systems Engineering Laboratory (WISELAB). He holds one Chinese patent and ten U.S. patents. He was a Guest Editor for a special issue on wireless networks for the International Journal of Communication Systems. His research interests include wireless communications and networks, spread spectrum and CDMA, antenna arrays and beamforming, SDR, and digital signal processing for wireless systems.

Dr. Yao is an Associate Editor of IEEE COMMUNICATIONS LETTERS and *IEEE Transactions on Vehicular Technology*, and an Editor for *IEEE Transactions on Wireless Communications*.

Harry Heffes (M'66-SM'82-F'90) received his B.E.E. degree from the City College of New York, New York, N.Y., in 1962, and the M.E.E. degree and the Ph.D. degree in Electrical Engineering from New York University, Bronx, N.Y., in 1964 and 1968 respectively. He joined the Department of Electrical Engineering and Computer Science at Stevens Institute of Technology in 1990. He assumed the faculty position of Professor of Electrical Engineering and Computer Science after a twenty eight year career at AT&T Bell Laboratories. In his early years at Bell Labs he worked on the Apollo Lunar Landing Program, where he applied modern control and estimation theory to problems relating to guidance, navigation, tracking and trajectory optimization. More recently his primary concern has been with the modeling, analysis, and overload control of telecommunication systems and services. He is the author of over twenty five papers in a broad range of areas including voice and data communication networks, overload control for distributed switching systems, queuing and teletraffic theory and applications, as well as computer performance modeling and analysis.

Dr. Heffes received the Bell Labs Distinguished Technical Staff Award in 1983 and, for his work on modeling packetized voice traffic, he was awarded the IEEE Communication Society's S.O. Rice Prize in 1986. He is a fellow the IEEE for his contributions to teletraffic theory and applications. He has served as a United States Delegate to the International Teletraffic Congress and as an Associate Editor for *Networks: An International Journal*. He is a Member of Eta Kappa Nu, Tau Beta Pi, American Men and Women of Science and is listed in Who's Who in America.