A Learning Automata Based Dynamic Guard Channel Scheme

Hamid Beigy and Mohammadreza Meybodi

Soft Computing Laboratory
Computer Engineering Department
Amirkabir University of Technology
Tehran, Iran
{beigy, meybodi}@ce.aku.ac.ir

Abstract. Dropping probability of handoff calls and blocking probability of new calls are two important QoS measures for cellular networks. Call admission policies, such as fractional guard channel and uniform fractional guard channel policies are used to maintain the pre-specified level of QoS. Since the parameters of network traffics are unknown and time varying, the optimal number of guard channels is not known and varies with time. In this paper, we introduce a new dynamic guard channel policy, which adapts the number of guard channels in a cell based on the current estimate of dropping probability of handoff calls. The proposed algorithm minimizes blocking probability of new calls subject to the constraint on the dropping probability of handoff calls. In the proposed policy, a learning automaton is used to find the optimal number of guard channels. The proposed algorithm doesn't need any a priori information about input traffic. The simulation results show that performance of this algorithm is close to the performance of guard channel policy for which we need to know all traffic parameters in advance. Two advantages of the proposed policy are that it is fully autonomous and adaptive. The first advantage implies that, the proposed policy does not require any exchange of information between the neighboring cells and hence the network overheads due to the information exchange will be zero. The second one implies that, the proposed policy does not need any priori information about input traffic and the traffic may vary.

1 Introduction

With increasing popularity of mobile computing, demand for channels is on the rise. Since number of allocated channels for this purpose is limited, the cellular and micro cellular networks are introduced, in which the service area is partitioned into regions called cells. Introduction of micro cellular networks leads to improvement of network capacity but increases the expected rate of handoff. When a mobile host moves across the cell boundary, handoff is required. If an idle channel is available in the destination cell, then the handoff call is resumed; otherwise the handoff call is dropped. Dropping probability of handoff calls (B_h) and blocking probability of new calls (B_n) are important quality of service (QoS)

M.H. Shafazand and A M. Tjoa (Eds.): EurAsia-ICT 2002, LNCS 2510, pp. 643–650, 2002.

measures of the cellular networks. Since the disconnection in the middle of a call is highly undesirable, B_h is more serious than B_n. In order to control B_n and B_h, *call admission control* (CAC) policies are introduced. The call admission policies determine whether a new call should be admitted or blocked. Both B_n and B_h are affected by call admission control policies. The simplest CAC policy is called *guard channel* policy (GC) [1]. Suppose that the given cell has C full duplex channels. The guard channel policy reserves a subset of channels, called *guard channels*, allocated to a cell for sole use of handoff calls (say $C - T$ channels). Whenever the channel occupancy exceeds the certain threshold T, the guard channel policy rejects new calls until the channel occupancy goes below T. The guard channel policy accepts handoff calls as long as channels are available. As the number of guard channels increased, B_h will be reduced while B_n will be increased [2]. It has been shown that there is an optimal threshold T^* in which B_n is minimized subject to the hard constraint on B_h [3]. Algorithms for finding the optimal number of guard channels are given in [3,4]. These algorithms assume that the input traffic is a stationary process with known parameters. The GC policy reserves an integral number of guard channels for handoff calls. In order to have more control on B_n and B_h, *limited fractional guard channel* (LFG) policy is introduced, which reserves a non-integral number of guard channels [3]. It has been shown that there is an optimal threshold T^* and an optimal value of π^* for which B_n is minimized subject to the hard constraint on B_h [3]. An algorithm for finding such optimal parameters is given in [3]. Since the input traffic is not a stationary process and its parameters are unknown a priori, the optimal number of guard channels is different for different traffic. In such cases the *dynamic guard channel* policy can be used. In dynamic guard channel policy, the number of guard channels varies during the operation of the cellular network.

Learning automaton (LA) is a reinforcement learning technique and has been used successfully in many applications such as telephone and data network routing [5,6], solving NP-Complete problems [7,8,9,10] and capacity assignment [11], to mention a few. In this paper, we propose an adaptive and autonomous call admission control algorithm, which uses LA. This algorithm uses only the current channel occupancy of the given cell and dynamically adjusts the number of guard channels. The proposed algorithm minimizes the blocking probability of new calls subject to the constraint on the dropping probability of handoff calls. Since the learning automaton starts its learning without any priori knowledge about its environment, the proposed algorithm does not need any a priori information about input traffic. One of the most important advantage of the proposed algorithm is that no status information will be exchanged between neighboring cells. The exchange of such status information increase the performance of the proposed algorithm. The simulation results show that the performance of this algorithm are near to performance of GC policy that knows all traffic parameters.

The rest of this paper is organized as follows: The section 2 presents the performance parameters of guard channel policy. The LA briefly is given in section 3. The proposed LA based dynamic guard channel policy is presented in section 4. The computer simulations is given in section 5 and section 6 concludes the paper.

2 The Blocking Performance of Guard Channel Policy

The blocking performance of guard channel policy is computed based on the following assumptions.

1. The arrival process of new and handoff calls is poisson process with rate λ_n and λ_h, respectively. Let $\lambda = \lambda_n + \lambda_h$.
2. The call holding time for both types of calls are exponentially distributed with mean μ^{-1}.
3. The time interval between two calls from a mobile host is much greater than the mean call holding time.
4. Only mobile to fixed calls are considered.
5. The network is homogenous.

The above first three assumptions have been found to be reasonable as long as the number of mobile hosts in a cell is much greater than the number of channels allocated to that cell. The fourth assumption makes our analysis easier and the fifth one lets us to examine the performance of a single network cell in isolation. Suppose that the given cell has a limited number of full duplex channels, C, in its channel pool. We define the state of a particular cell at time t to be the number of busy channels in that cell, which is represented by $c(t)$. The $\{c(t)|t \geq 0\}$ is a continuous-time Markov chain (birth-death process) with states $0, 1, \ldots, C$. The state transition rate diagram of a cell with C full duplex channels and dynamic guard channel policy is shown in figure 1.

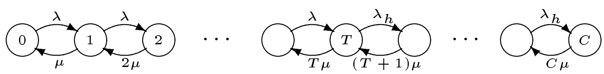

Fig. 1. Markov chain model of cell

Because of the structure of the Markov chain, we can easily write down the steady-state balance equations. Define the steady state probability $P_n = \lim_{t\to\infty} \text{Prob}[c(t) = n]$ for state $n = 0, 1, \ldots, C$. Then, the following expression can be derived for P_n $(n = 0, 1, \ldots, C)$.

$$P_n = \begin{cases} \frac{\rho^k}{k!} P_0 & k = 0, 1, \ldots, T-1 \\ \\ \frac{\rho^k \alpha^{k-T}}{k!} P_0 & k = T, \ldots, C, \end{cases} \tag{1}$$

where $\rho = \lambda/\mu$, $\alpha = \lambda_h/\lambda$ and P_0 is the probability that all channels are free and given by the following expression.

$$P_0 = \left[\sum_{k=0}^{T-1} \frac{\rho^k}{k!} + \sum_{k=T}^{C} \frac{\rho^k \alpha^{k-T}}{k!} \right]^{-1} \tag{2}$$

Thus, dropping probability of handoff calls, $B_h(C,T)$, ie equal to $B_h(C,T) = \frac{\rho^C \alpha^{C-T}}{C!}$ and blocking probability of new calls is equal to $B_n(C,T) = \sum_{k=T}^{C} P_k$.

The objective of call admission control policies is to find a T^* that minimizes the $B_n(C,T^*)$ given the constraint $B_h(C,T^*) \leq p_h$. The value of p_h specifies by the quality of service of the network. In order to find the optimal value of T^*, in [3] a binary search and in [4] a linear search algorithms are given. These algorithms assumes that the all parameters of input traffic are known in advance.

3 Learning Automata

The automata approach to learning involves the determination of an optimal action from a set of allowable actions. An automaton can be regarded as an abstract object that has finite number of actions. It selects an action from its finite set of actions. This action is applied to a random environment. The random environment evaluates the applied action and gives a grade to the selected action of automata. The response from environment (i.e. grade of action) is used by automata to select its next action. By continuing this process, the automaton learns to select an action with best grade. The learning algorithm used by automata to determine the selection of next action from the response of environment. An automaton acting in an unknown random environment and improves its performance in some specified manner, is referred to as *learning automaton* (LA). Learning automata can be classified into two main families: *fixed structure learning automata* and *variable structure learning automata* [12].

Variable structure learning automata are represented by triple $< \beta, \alpha, T >$, where β is a set of inputs actions, α is a set of actions, and T is learning algorithm. The learning algorithm is a recurrence relation and is used to modify the state probability vector. It is evident that the crucial factor affecting the performance of the variable structure learning automata, is learning algorithm. Various learning algorithms have been reported in the literature. Let α_i be the action chosen at time k as a sample realization from probability distribution $p(k)$. In linear reward-ϵpenalty algorithm ($L_{R-\epsilon P}$) scheme the recurrence equation for updating p is defined as

$$p_j(k+1) = \begin{cases} p_j(k) + a \times [1 - p_j(k)] \text{ if } \quad i = j \\ \\ p_j(k) - a \times p_j(k) \qquad \text{if} \quad i \neq j \end{cases} \quad \text{if} \quad \beta(k) = 0 \tag{3}$$

$$p_j(k+1) = \begin{cases} p_j(k) \times (1-b) \qquad \text{if} \quad i = j \\ \\ \frac{b}{r-1} + p_j(k)(1-b) \text{ if } \quad i \neq j \end{cases} \quad \text{if} \quad \beta(k) = 1 \tag{4}$$

The parameters $0 < a < 1$ and $0 < b \ll a$ represent *step lengths* and r is the number of actions for learning automata. The a and b determine the amount of increase and decreases of the action probabilities, respectively. If the a equals to b the recurrence equations (3) and (4) is called *linear reward penalty*(L_{R-P}) algorithm.

```
if (NEW CALL) then
    set g ← LA.action ()
    if (c(t) < C − g )then
        accept call
        if (B̂_h < p_h ) then
            reward action g
        else
            penalize action g
        end if
    else
        reject call
        if (B̂_h < p_h ) then
            penalize action g
        else
            reward action g
        end if
    end if
end if
```

Fig. 2. LA based dynamic guard channel algorithm

4 LA Based Dynamic Guard Channel Policy

In this section, we introduce a new LA based algorithm (figure 2) to determine number of guard channels when the parameters λ_n, λ_h, and μ are unknown and possibly time varying. In this algorithm, LA is used to adjust number of guard channels. Assume that the cell has C full duplex channels. Let the number of guard channels at time instant t denoted by $g(t)$ is in interval $g(t) \in [g_{\min}, g_{\max}]$, where $0 \leq g_{\min} \leq g_{\max} \leq C$. In the proposed algorithm, each base station has a LA with $g_{\max} - g_{\min} + 1$ actions, where action α_i denotes that the base station must use $g(t) = g_{\min} + \alpha_i - 1$ guard channels. The proposed algorithm can be described as follows. When a handoff call arrives at the given cell and a channel is available, then the call is accepted; otherwise it is dropped. When a new call arrives at the given cell, LA associated to the cell selects one of its actions, say α_i. If the cell has at least $g_{\min} + \alpha_i - 1$ free channels, then the incoming call is accepted; otherwise it is blocked. Then the base station computes the current estimate of dropping probability of handoff calls $(\hat{B}_h)$ and then compare this quantity with the specified level of QoS (p_h). If the incoming new call is accepted and the current value of $(\hat{B}_h)$ is less than p_h then action α_i is rewarded; otherwise penalized. If the incoming new call is blocked and the current value of $(\hat{B}_h)$ is greater than p_h then the action α_i is rewarded; otherwise the action α_i is penalized. The comparison of current estimate of dropping probability of handoff calls and the specified level of QoS (p_h) is done to guarantee the specific level of QoS. The proposed algorithm requires less resources (bandwidth of the wired-line network) than other distributed call admission algorithm for which the status of all neighboring cells are needed for determination of guard channels. In other distributed call admission algorithms, status information must

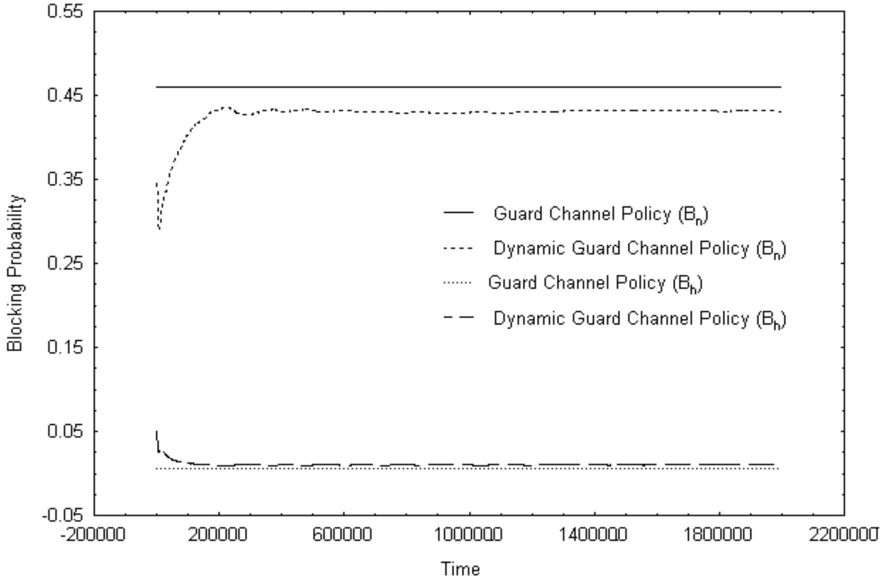

Fig. 3. The comparison of guard channel policy and dynamic guard channel policy

be exchanged between neighboring cells in the case of arrival of a call, departure of a call, and handoff of a call. However, the exchange of status information can be used to speed up the convergence of the proposed algorithm, which results an improvement of the proposed algorithm. Since the learning automata begin their learning without a priori knowledge about its environment, the proposed algorithm does not require any information about input traffic. Even though the priori information about input traffic is not needed by the algorithm, availability of such information may be used to find a better learning algorithm in order to choose a better learning algorithm for adaptation of traffic parameters. The use of a priori information in the proposed algorithm needs to be investigated. The proposed algorithm at the beginning does not perform well but as it proceeds, the performance of the algorithm approaches to its optimal performance. Initially, the proposed guard channels randomly.

5 Simulation Results

In this section, we compare performance of the guard channel [1], the limited fractional guard channel [3], and the dynamic guard channel algorithms proposed in this paper. The results of simulations are summarized in table 1. The simulation is based on the single cell of homogenous cellular network system. In such network, each cell has 8 full duplex channels ($C = 8$). In the simulations, new call arrival rate is fixed to 30 calls per minute ($\lambda_n = 30$), channel holding time is set to 6 seconds ($\mu^{-1} = 6$), and the handoff call traffic is varied between 2 calls per minute to 20 calls per minute. The results listed in table 1 are obtained by averaging 10 runs from 2,000,000 seconds simulation of each algorithm. The objective is to minimize the blocking probability of new calls subject to the constraint that the dropping probability of handoff calls is less than 0.01. The optimal number of guard channels for guard channel policy is obtained by algorithm given in [4] and the optimal parameters of limited fractional guard channel policy is obtained by algorithm given in [3].

Table 1. The simulation results of the LA base dynamic guard channel policy

		GC		LFG		DGC	
Case	λ_h	B_n	B_h	B_n	B_h	B_n	B_h
1	2	0.063507	0.001525	0.031609	0.023283	0.053433	0.010619
2	4	0.077080	0.003538	0.051414	0.020675	0.080966	0.010039
3	6	0.091013	0.005923	0.071632	0.018707	0.125500	0.009964
4	8	0.105002	0.008380	0.092138	0.016706	0.154861	0.010031
5	10	0.120260	0.011877	0.114445	0.015572	0.207490	0.010067
6	12	0.231559	0.004309	0.147902	0.014044	0.245842	0.010017
7	14	0.255346	0.005975	0.204217	0.012675	0.290619	0.009960
8	16	0.275489	0.007999	0.250642	0.011554	0.331478	0.009983
9	18	0.296834	0.010518	0.294441	0.010877	0.377334	0.009953
10	20	0.459183	0.006081	0.384157	0.010182	0.427894	0.010005

By inspecting table 1, it is evident that the performance of dynamic guard channel policy is close to the performance of guard channel policy. One reason for the difference in performances of the guard channel policy and the proposed policy is due to the fact that transient behavior of the proposed algorithm. Since, the performance parameters (the blocking probability of new calls and the dropping probability of handoff calls) in the early stages of simulation are far from their desire value, they affect the long-time calculation of the performance parameters. However, such effect can be removed by excluding the transient behaviors of the proposed algorithm, which is shown in figure 3. Figure 3 shows the evolution of the performance parameters for the guard channel policy and the proposed dynamic guard channel policy. The traffic parameters used for figure 3 corresponds to case 10 in table 1. By carefully inspecting figure 3 and ignoring the transient behavior of the proposed algorithm, it can be concluded that the dropping probability of handoff calls approaches its prescribed value (p_h), while the blocking probability of new calls is less than the corresponding performance parameter of guard channel policy.

6 Conclusions

In this paper, a dynamic guard channel policy based on learning automata is given. The proposed algorithm adapts the number of guard channels in a cell using current estimate of dropping probability of handoff calls. This algorithm minimizes blocking probability of new calls subject to the constraint on dropping probability of handoff calls. The simulation results show that the performance of this algorithm is very close to the performance of guard channel policy that knows all traffic parameters in advance. The proposed policy has three advantages: 1) doesn't require any exchange of information between the neighboring cells leading to less network overheads. 2) doesn't need any a priori information about the input traffic. 3) the algorithms works for time varying traffics.

References

1. D. Hong and S. Rappaport, "Traffic Modelling and Performance Analysis for Cellular Mobile Radio Telephone Systems with Priotrized and Nonpriotorized Handoffs Procedure," *IEEE Transactions on Vehicular Technology*, vol. 35, pp. 77–92, Aug. 1986.
2. S. Oh and D. Tcha, "Priotrized Channel Assignment in a Cellular Radio Network," *IEEE Transactions on Communications*, vol. 40, pp. 1259–1269, July 1992.
3. R. Ramjee, D. Towsley, and R. Nagarajan, "On Optimal Call Admission Control in Cellular Networks," *Wireless Networks*, vol. 3, pp. 29–41, 1997.
4. G. Haring, R. Marie, R. Puigjaner, and K. Trivedi, "Loss Formulas and Their Application to Optimization for Cellular Networks," *IEEE Transactions on Vehicular Technology*, vol. 50, pp. 664–673, May 2001.
5. P. R. Srikantakumar and K. S. Narendra, "A Learning Model for Routing in Telephone Networks," *SIAM Journal of Control and Optimization*, vol. 20, pp. 34–57, Jan. 1982.
6. O. V. Nedzelnitsky and K. S. Narendra, "Nonstationary Models of Learning Automata Routing in Data Communication Networks," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-17, pp. 1004–1015, Nov. 1987.
7. B. J. Oommen and E. V. de St. Croix, "Graph Partitioning Using Learning Automata," *IEEE Transactions on Commputers*, vol. 45, pp. 195–208, Feb. 1996.
8. H. Beigy and M. R. Meybodi, "Backpropagation Algorithm Adaptation Parameters using Learning Automata," *International Journal of Neural Systems*, vol. 11, no. 3, pp. 219–228, 2001.
9. M. R. Meybodi and H. Beigy, "New Class of Learning Automata Based Schemes for Adaptation of Backpropagation Algorithm Parameters," *International Journal of Neural Systems*, vol. 12, pp. 45–68, Feb. 2002.
10. M. R. Meybodi and H. Beigy, "A Note on Learning Automata Based Schemes for Adaptation of BP Parameters ," *Accepted for Publication in the Journal of Neuro Computing, To Appear.*
11. B. J. Oommen and T. D. Roberts, "Continuous Learning Automata Solutions to the Capacity Assignment Problem," *IEEE Transactions on Commputers*, vol. 49, pp. 608–620, June 2000.
12. K. S. Narendra and K. S. Thathachar, *Learning Automata: An Introduction.* New York: Printice-Hall, 1989.