# A DISTRIBUTED REINFORCEMENT LEARNING APPROACH TO MAXIMIZE RESOURCE UTILIZATION AND CONTROL HANDOVER DROPPING IN MULTIMEDIA WIRELESS NETWORKS

**Eftychia Alexandri[1], Georges Martinez[1], Djamal Zeghlache[2]**

[1] Motorola Labs, Espace Technologique de Saint-Aubin, 91193 Gif-sur-Yvette Cedex, France, Eftychia.Alexandri@int-evry.fr
[2] Institut National des Télécommunications, 9 rue Charles Fourier, 91011 Evry Cedex, France

**Abstract** - A new scheme to maximize resource utilization in a cellular network while respecting constraints on handover dropping probability is proposed and analyzed. The constraints are set for each traffic class separately and have to be respected by the network independently of the area in a localized manner. The problem is formulated as a Markov Decision Process (MDP) and solved by making use of the model-free simulation-based Q-learning algorithm that runs at each cell. Integration of the handover limit in the model is achieved by observing which of the new call arrivals, at a particular state of the system, are mostly responsible for violation of the handover dropping limit. Through trial and error, the algorithm proceeds to the statistical elimination of new admissions in the system, those causing excessive dropping. Results obtained via the proposed Reinforcement Learning (RL) based approach are compared with a resource allocation that takes into consideration heterogeneous and unevenly distributed traffic over the geographical area under consideration. For the scenarios examined, comparable results and performance are observed with an advantage for RL in blocking and utilization.

## I. INTRODUCTION

In the near future, wireless cellular networks will be used to carry multimedia traffic. These heterogeneous services needs, combined with user mobility, require cost-efficient solutions. From the network provider perspective the interest is in schemes generating revenue through maximal resource utilization while achieving QoS targets. For this paper the focus is on resource utilization, call blocking and call dropping.

Handover dropping control has been typically achieved through trunk reservation in each cell. An a priori resource reservation in the form of guard channels provides priority to handover calls over new calls. Proposals include fixed and dynamic reservation of resources as well as optimal and heuristic solutions ([1]-[3]).

The current study integrates the two quality metrics in an optimization process that maximizes resource utilization while keeping the handover dropping probability per service below a predefined limit. The problem is formulated as a sequential decision one, and solved by making use of the model-free approach of Reinforcement Learning. Dropping probability constraints can be different per service class and set to a different value per region. The proposed scheme is then compared to a method based on handover-related resource allocation in a multimedia context. In opposition to the other method, RL requires no exchange of control information among neighboring cells and no explicit knowledge of the transition model. RL relies on exploration of the state space to converge towards a policy that provides maximal resource utilization. We make use of this exploration to integrate a new mechanism that identifies which among the new call admissions induce excessive call/session dropping. We subsequently restrain from admitting these new calls; thus we prevent handover dropping degradation.

Section II presents the model for the wireless network and the traffic and mobility models. In section III the optimization is cast into a sequential decision problem using the Markov Decision Process (MDP) framework. Section IV describes the Reinforcement Learning Q-algorithm used to solve the problem in a model-free manner. This approach offers easy extensibility. Section V introduces the original idea to keep handover dropping from exceeding predefined limits by exercising selective admission control on new calls. Section VI briefly highlights the ExpectedMax algorithm used for performance comparisons. Results are reported and discussed in section VII.

## II. SYSTEM MODEL

Consider a wireless cellular network organized into geographical regions called cells. Each cell has a number of resource or bandwidth units (BWUs). Upon arrival of a service demand, the cell allocates for a certain time interval part of its resources at call or session establishment. A service s is characterized by the necessary number of bandwidth units $b_s$ needed to render the service. Each service, in each cell, is characterized by an arrival rate, a residence time (or cell dwell time - time that the user of the service spends in the cell before moving on to one of the neighboring cells) and a total call duration. Assuming independent call/session arrivals, the arrivals form a Poisson process. The call duration and the cell residence times are assumed exponentially distributed. The mean values can vary on a cell and service basis. For each cell i, a handover probability $p_{ij}$ is associated to each of its neighboring cells j.

Whenever a service flow currently in cell i wishes to move to another cell, the next cell j is picked with probability $p_{ij}$.

## III. PROBLEM FORMULATION

### *A. Introduction*

The problem that we are addressing can be regarded as a large-scale, constrained optimization problem embedded in a stochastic environment. Learning can provide answers to this type of problems. This is not a one step decision problem where one could assume that the decision at each call arrival or at each handover demand is independent of previous or future decisions. Past and future decisions have an impact on system state. The problem is thus perceived as a sequential decision problem. Markov Decision Processes (MDP) provide a framework for solving sequential decision problems, where the state transitions have the memoryless (Markov) property. Reinforcement Learning (RL) methods are goal-directed learning from interaction, where one learns what to do (mapping from situations to actions) in order to maximize a numerical reward signal. RL can be used to solve problems formulated as MDPs without the need for an explicit model (transition probabilities). Learning can be conducted either off-line (prior to its application on an operating system) or on-line (while the system is operating). These features make RL attractive for solving call admission control and resource allocation problems in wireless mobile environments.

### *B. MDP general framework*

Markov Decision Processes consist of :

- a set of states S,
- a set of actions A,
- an immediate reward function

  $R : SxA \rightarrow \Re$ ($\Re$ is the set of real numbers),
- a state transition function

  $T : SxA \rightarrow \Pi(S)$,

where $\Pi(S)$ is the probability distribution over set S. T(s, a, s') is the probability of making a transition from state s to state s' upon choosing action a. The transitions depend on the last state and action (Markovian property).

The objective is to find a policy $\pi$, which is a mapping from states to actions ($\pi$:SxA), that achieves in the long run the goal of maximizing the discounted sum of the immediate rewards received over an infinite horizon.

$$\max_{\pi} E\left(\sum_{t=0}^{\infty} \gamma^t r_t\right) \qquad (1)$$

Variable $r_t$ is the reward the system gets at time t upon taking action a in state s. The discount factor $0 \leq \gamma < 1$ indicates the impact or not of future actions and their associated rewards (for $\gamma=0$ future rewards have no impact on the state value, while for $\gamma$ close to 1 future actions have a very strong impact).

### *C. Problem Formulation*

The state of each cell c at time t consists of the configuration of the cell and the event happening at time t, upon which we have to make a decision. The cell configuration is given by the number of flows per service s that are being served by the cell, and also by the dropping probability for each of the services. In order to have a finite number of states, we represent the dropping probability within a state as an interval. The size of these contiguous intervals is not fixed. The smallest intervals are used to represent values from 0 up to slightly higher than the dropping probability limit per service $P_{drop\text{-}max,s,c}$. A "forbidden" interval, where we do not want our system to be found, is used from a little more than $P_{drop\text{-}max,s,c}$ up to probability 1. An event corresponds to the arrival of a new flow of service s or the arrival of a handed over flow of service s from one of the neighbouring cells. Thus, for a cell c of total capacity of $C_c$ bandwidth units, and a total of S services, the defined state space is finite.

Possible actions are :

- admitting or refusing access to the service flow when there are enough resources in the cell,
- rejecting the flow when the available resources at the cell level are not sufficient.

The immediate reward $r_{t+1}$ received by the system at decision time (t+1) is the product of the bandwidth, (occupied by the service admitted at decision time t) by the time interval between decision epochs (t+1) and t. If no allocation has taken place at decision time t, the immediate reward $r_{t+1}$ is zero. As no decision is taken for service flows whose call duration comes to an end, there is no associated event for these flows. Resources are simply freed in this case.

## IV. REINFORCEMENT LEARNING MECHANISM

Q-learning is an RL algorithm which provides a solution to an MDP problem without the need for explicit knowledge of transition probability and reward functions. It accomplishes that by iteratively approximating the optimal value of each current state based on the value of the next state upon applying the optimal action associated with the current state. It can approximate the optimal policy with probability one, provided that the state space and the action space are finite. Q-learning requires an explicit enumeration of all state-action pairs (s, a) that are encountered in the system. A numerical value (a reward) is associated with each pair (s, a). Nevertheless, for more realistic problems, with large state spaces, where the use of such an exhaustive enumeration gets impractical, approximations have been successfully used [4].

To understand how Q-learning functions let us define the total expected discounted return V(s) over an infinite time horizon, if we consider that we start at state s. The optimal value V*(s) of state s is defined as the total expected discounted return over an infinite time horizon, if we consider that we start at state s and follow the optimal policy:

$$V^*(s) = \max_{\pi} E\left(\sum_{t=0}^{\infty} \gamma^t r_t\right) \quad (2)$$

Expressing the state value as a function of the immediate reward resulting from action a in state s leading to state s':

$$V^*(s) = \max_a \left(R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') V^*(s')\right) \forall s \in S \quad (3)$$

Let us define $Q^*(s,a)$ as the expected discounted reward/reinforcement of taking action a in state s. Continuing the selection of optimal actions, it follows that:

$$V^*(s) = \max_a Q^*(s,a) \quad (4)$$

The previous equation giving V*(s) becomes:

$$Q^*(s,a) = R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') \max_{a'} Q^*(s',a') \quad (5)$$

By updating the Q-values according to the following iteration, on a transition from state s to s' following action a, the Q-values for all potential pairs (s, a) are obtained:

$$Q(s,a) = Q(s,a) + \alpha(r + \gamma \max_a Q^*(s',a') - Q(s,a)) \quad (6)$$

where $\alpha$ is the step size/ learning rate, which for $\sum_{k=1}^{\infty} \alpha_k = \infty$ and $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$ guarantees convergence in a stationary environment.

Note that the previous equation is independent of both R(s, a) and T(s, a, s'), thus eliminating the need to explicitly calculate the transition probabilities and the reward function. In order for the Q-learning algorithm to perform well, all potentially important state-action (s, a) pairs have to be visited/explored.

In our scheme, we use one Q-learning per cell. We set the values of $\alpha$ to 0.01 and $\gamma$ to 0.99.

## V. HANDOVER VS BLOCKING TRADEOFF

In order to keep the handover dropping probability $P_{drop,s,c}$ below the upper limit set per service s, in cell c, the handover failure of every service in each cell is continuously monitored. Furthermore, for each new service flow admitted in each cell, the state s is stored at the time the admission decision a for the new flow is taken.

If $P_{drop,s,c}$ becomes higher than the predefined limit $P_{drop-max,s,c}$, the list of all new flows is examined, independently of the service to which they belong, and the couple (s, a) of the state s and the admission decision/action a, at which they were actually admitted into the system, is marked as potentially "suspect". This initial marking, following the handover dropping violation in the cell, is done via a counter $cm_{(s,\ a)}$. RL visits all potential states, in the exploration phase, and we take advantage of this fact to proceed to the statistical elimination of cases where a new admitted flow in the cell leads to a subsequent handover dropping probability violation. In order to know how often the couple (s, a) is associated with a subsequent undesired dropping probability, a second counter $cv_{(s,\ a)}$ keeps track of how many times (s, a) has been visited in total.

By examining the ratio $rc = cm_{(s,\ a)}/cv_{(s,\ a)}$, for all (s, a) in the Q-table and comparing to a threshold $rc_{thr}$, the algorithm prevents the system from taking an action a in a state s, when $rc > rc_{thr}$.

## VI. PRIOR DYNAMIC RESOURCE ALLOCATION DURING HANDOVER (EXPECTEDMAX)

In this section we present the prior ExpectedMax algorithm for dynamic resource allocation during handover, which has been introduced in [1]. The objective in this algorithm is to minimize the handover dropping probability for every service in a wireless cellular network. Different cells can have different service mixes, different capacities, and can be characterized by different mobility patterns.

Upon a new arrival of service s, asking for $b_s$ resources at cell c, ExpectedMax estimates, based on its knowledge of service traffic and mobility characteristics, how much time the new flow is expected to stay in the cell, once its admission is decided. Cell c then calculates, within this estimated time interval $\Delta t$, how many flows, for each service, are expected to move out of the cell or come to an end. At the same time, cell c requests from all of its neighbouring cells to communicate the number of flows, per service, which are expected to handover to cell c within the time interval $\Delta t$. Based on the information communicated from neighbour cells, and the information obtain by local calculations, cell c computes the number of probable sequences of incoming and outgoing events, for each of the services. The maximum number of resources, which will be needed for each of the sequences of incoming and outgoing events, is thus obtained. Attributing to each of the sequences the same probability of occurrence, the expected maximum (ExpectedMax) bandwidth requirement $Y_s$ per service s is deduced. The algorithm sums up these values for all

services, $\sum_{s=1}^{S} Y_s$, and decides to accept the new flow in cell c only if the available bandwidth in c exceeds $\sum_{s=1}^{S} Y_s + b_s$.

ExpectedMax yields better results than other fixed and dynamic resource allocation algorithms, and is found to adapt to different network conditions and traffic patterns. Nevertheless, the method requires heavy information exchange between neighboring cells and a lot of computation within cells. Each cell is supposed to have knowledge of the explicit model of traffic and mobility patterns for each service.

The RL approach proposed in this paper is compared to ExpectedMax that serves as a reference to assess performance. Even if the two methods differ in terms of exact objectives, the selection of ExpectedMax for comparison is based on its good performance and its ability to minimize handover failures and adapt well to varying environment and traffic conditions.

In the following we investigate whether ExpectedMax gives comparable results for resource utilisation, and how close it might be to design parameters of upper limit handover dropping probability, set by the network operator, as traffic load changes.

## VII. EVALUATION

The network topology of [1] with 19 cells is used as the base scenario. It consists of a down-town center cell c0 surrounded by a first tier of 6 city cells, c1 to c6, and a second tier of 12 suburban cells, c7 to c18.

The available resource/capacity differs between cells. The capacity for the down-town cell is 15 BWUs, 10 for city cells while 6 BWUs are available per suburban cell. Mobility among cells correspond to users moving from suburbs more towards the city, and from there more towards the city center.

Two services are considered with service S1 requiring 1 BWU and corresponding to a voice service. Service S2 has a need of 3 BWUs and corresponds to a data application. These needs correspond to effective bandwidths. For S1, call duration has a mean value of 150s and a cell residence time of 100s. For S2, mean call duration is 600s and cell residence time is 500s. Results comparing handover dropping probabilities, blocking probabilities and average resource utilization, between the proposed approach and ExpectedMax, are provided in the following figures. Service S1 is characterized by a mean arrival rate of 0.005.

Handover dropping probability is lower for the ExpectedMax strategy as depicted in figures 1a through 1c. This is expected since ExpectedMax aims at minimizing handover failures. RL maintains handover dropping probability below the predefined thresholds of 5% for service S1 and 10% for service S2. In lower traffic loads, our RL approach does better than simply respecting these target constraints.

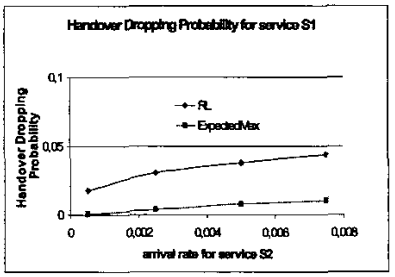


Fig 1a. Dropping Probability for service S1. Limit set at 0.05 for RL

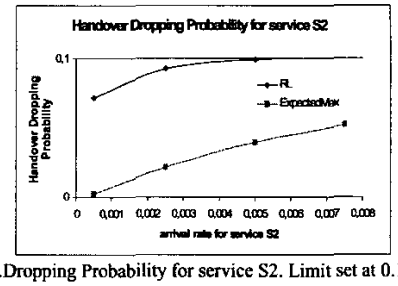


Fig 1b.Dropping Probability for service S2. Limit set at 0.1 for RL

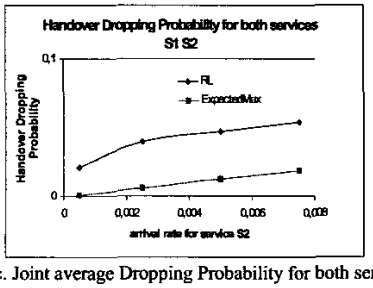


Fig 1c. Joint average Dropping Probability for both services

In figures 2a through 3c, RL achieves lower blocking probabilities and higher resource utilization.

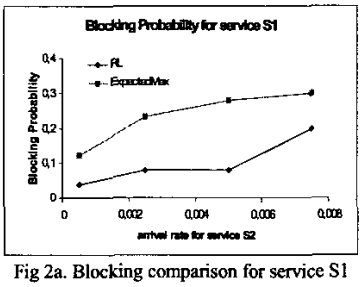


Fig 2a. Blocking comparison for service S1

RL exhibits fairly stable resource utilization efficiency. When traffic load is high, resource utilization decreases due to the admission control embedded in the RL algorithm that blocks more users to satisfy the dropping limits. At such loads the performance of both algorithms become fairly close. The ability of RL to achieve equivalent and for some measures better performance is an indication of an ability to adapt well to traffic variations in the network. RL has the

clear advantage of not requiring any exchange of control information among neighboring cells and not needing any explicit knowledge of the system transition model.

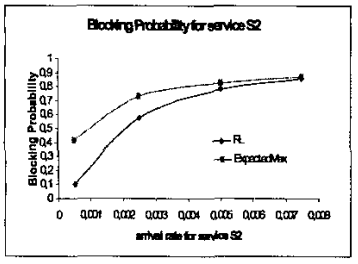


Fig 2b. Blocking comparison for service S2

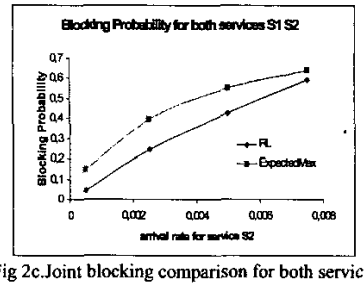


Fig 2c.Joint blocking comparison for both services

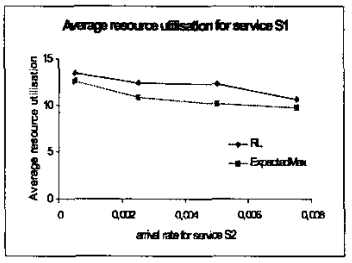


Fig 3a. Resource utilization for service S1

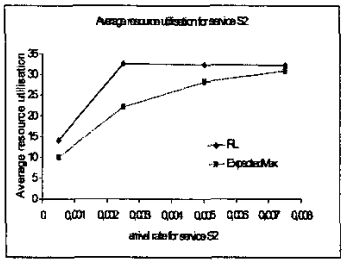


Fig 3b. Resource utilization for service S2

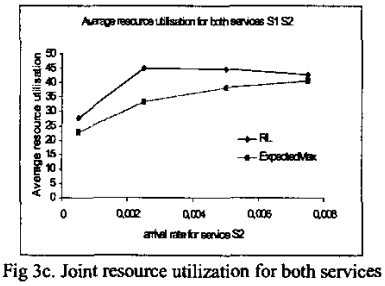


Fig 3c. Joint resource utilization for both services

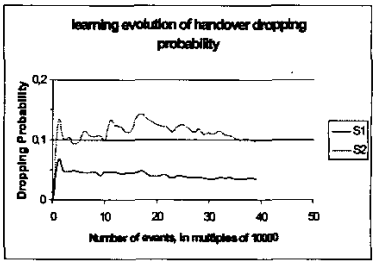


Fig 4. Learning of dropping probability limits for each service

Figure 4 depicts the evolution in time of the dropping probability for each of the two services separately, for the entire network. After exploration of the state space, the proposed distributed algorithm converges to dropping probability values respecting the limits of 0.05 for S1 and 0.10 for S2.

## VIII. CONCLUSIONS

This paper demonstrates the feasibility of using Reinforcement Learning to maximize resource utilization in a cellular network, in the presence of multimedia traffic while respecting constraints on handover dropping probability. The problem is formulated as a Markov decision process and solved by making use of the model-free, simulation-based Q-learning algorithm, which runs at each cell. In order to respect handover dropping limits, a novel approach that consists in observing which of the new call arrivals, at a particular state, cause handover dropping is introduced. The algorithm proceeds to the statistical rejection of call establishment requests generating increased handover failures. The algorithm achieves performance comparable to other classical methods but does not require the explicit knowledge of state transition probabilities.

## REFERENCES


[1] P. Ramanathan, K.M. Sivalingam, P. Agrawal, and S. Kishore, "Dynamic resource allocation schemes during handoff for mobile multimedia wireless networks", *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 7, July 1999

[2] Y.C. Ko, S.C. Park, C.Y. Chun, H.W. Lee, and C.H. Cho, "An adaptive QoS provisioning distributed call admission control using fuzzy logic control", *IEEE International Conference on Communications ICC 2001*

[3] E.S. El-Alfy, Y.D. Yao, and H Heffes, "A model-based Q-learning scheme for wireless channel allocation with prioritized handoff", *IEEE Global Telecommunications Conference Globecom 2001*

[4] R. S. Sutton, A. G. Barto, *Reinforcement Learning: An introduction*, MIT Press, Cambridge, MA, 1998