Comment [JL1]: General note – because this chapter contains a large number of abbreviations, please check that the abbreviations are correct for the English version

Chapter 11

Learning Techniques in a Mobile Network

11.1. Introduction

Because of the current evolution of society, people are increasingly on the move and need to communicate during their travels. This phenomenon has triggered greater demand and studies oriented towards the development of very sophisticated systems in order to respond to new user requirements. These requirements have indeed changed: if originally only voice was needed, wireless transmission demand providing reliable high definition sound, image and even high quality video communications has increasingly become popular with a large number of users. These users hope for mobility to be completely transparent in order to take advantage of performances similar to those from wired networks, despite the bandwidth greed of these new services.

Cellular systems are without a doubt those having experienced the strongest growth these last few years. The geographical zone served by a cellular network is divided into small surfaces called cells. Each of them is covered by a transmitter called “base station” (BS). Bandwidth in this type of network is divided into a separate group of radio channels defined by the access technique used. These channels can be used simultaneously as long as acceptable radio signal quality is maintained. This division can be done with different access techniques such as FDMA (*frequency-division multiple access*), TDMA (*time-division multiple access*), CDMA (*code-division multiple access*) or any combination of these methods [CAL 92, GIB 96, AGH01]. What is left now is to define the way in which these

Chapter written by Sidi-Mohammed SENOUCI.

channels are attributed to cells. There are two main methods: FCA (*fixed channel allocation*), where each cell has a specific number of channels and DCA (*dynamic channel assignment*) where all channels are grouped in a common pool (or group) and are dynamically assigned to cells [KAT 96]. In FCA or DCA type systems, a free channel not violating the constraint of channel reuse[1] is allocated to each user. However, when the user goes from one cell to another, he must request a new free channel in the destination cell. This event, called intercellular transfer or "handoff", must be transparent to the user. If the destination cell has no available channel, the call is disconnected. One of the great challenges for this type of network is managing this user mobility during a communication. In fact, the availability of radio resources for the duration of communication is not necessarily guaranteed and these users can experiment communication degradation or even break during intercellular transfer.

One of the major concerns during cellular network design is break probability decrease. In fact, from a user viewpoint, it is much more unpleasant than a connection failure. This is all the more important because cell size keeps decreasing in order to respond to cellular network growth, which considerably increases the number of intercellular transfers. Thus, since radio resource is a scarce resource, it is imperative to use it to the maximum, particularly in the case of a multiservice cellular network supporting several traffic classes where each one requires a different QoS level. For an administrator, it is always preferable to block a call from a lower priority service class (data, for example) and to accept another call with a higher priority service class (voice, for example). Consequently, a good call admission control (CAC) policy is certainly vital to maximize the usefulness of all these radio resources. To reach this objective, it is also necessary to find a good allocation method of all bandwidth available to all cells. These new mechanisms (CAC, dynamic resource allocation) must also handle frequent traffic conditions changes in cellular networks.

The objective of this chapter is to prove that it is possible to use techniques from the world of artificial intelligence (AI) and more specifically learning techniques in order to develop robust and efficient mechanisms to solve the problems encountered in cellular networks. These mechanisms must also exploit experience and knowledge which could be acquired during network operation.

In order to do this, we have developed new call admission control mechanisms in a cellular network considering both channel allocation diagrams: fixed (FCA) and dynamic (DCA). We have also developed a new dynamic resource allocation mechanism for choosing the best channel among all available channels in the

1 A channel can be used in many cells as long as the interference constraint is respected.

common pool, with the objective of maximizing the usage rate of all channels. These solutions are obtained by using the reinforcement learning Q-learning algorithm [WAT 89, WAT 92].

The rest of the chapter is organized as follows: section 11.2 briefly presents the notion of learning by emphasizing reinforcement learning and its application in telecommunications networks; section 11.3 presents a new call admission control method in cellular networks based on reinforcement learning; section 11.4 discusses a new dynamic radio resource allocation policy in cellular systems, also based on reinforcement learning; finally, section 11.5 concludes this chapter.

11.2. Learning

The term learning designates the capability to organize, develop and generalize knowledge for future use. It is the capability to take advantage of experience to improve problem resolution. Depending on the type of information available, two main approach categories can be observed. The first one, qualified as unsupervised learning, attempts to group objects into classes, relying on similarities. The second approach, i.e. supervised learning, is based on a learning group made up of objects where the class is already known.

11.2.1. *Unsupervised learning*

Unsupervised learning, also called learning from observation, consists of defining a classification from a group of objects or given situations. We use a mass of indistinct data and we wish to know if they have any group structure. The objective is to identify future data trends to be grouped into classes. This type of learning called clustering is found in automatic classification and in digital taxonomy. It searches for consistencies among a group of examples, without necessarily being guided by the use of acquired knowledge. It groups these examples in such a way that examples within one group are close enough and examples of different groups are different enough.

11.2.2. *Supervised learning*

In this type of learning, a teacher (or supervisor, hence the name supervised learning) provides either the action which should be executed, or an error gradient. In both cases, the teacher provides a controller with an indication of the action that it

should generate in order to improve its performance. The use of this approach presupposes the existence of an expert able to provide a group of examples, called learning base, which is made up of correct associated situations and actions. These examples must be representative of the task to accomplish.

One of the variations of supervised learning, in which a "critique" of the calculated response is provided to the network, is called reinforcement learning (RL). This algorithm variation, explained below, has appeared as the most adapted to solve problems related to cellular networks and treated in this chapter.

11.2.3. *Reinforcement learning*

Reinforcement learning (also called semi-supervised learning) is a variation of supervised learning [SUT 98]. In contrast with the supervised approach, the teacher agent in reinforcement learning has a role of evaluation and not instruction. It is generally called critique. The role of critique is to provide a measure indicating whether the action generated is appropriate or not. The objective is to program an agent with the help of a penalty/reward evaluation without having to specify how the task must be accomplished. In this context, we must indicate to the system what goal to reach and the system must learn, by a series of trials and errors (in interaction with the environment), how to reach the set goal.

The components of reinforcement learning are the "apprentice" agent, its environment and the task to carry out (see Figure 11.1). The interaction between agent and environment is continuous. On the one hand, the agent's decision process chooses the actions based on situations perceived from its environment. On the other hand, these situations are influenced by these actions. Each time the agent accomplishes an action, it receives a reward. This reward is a scalar value indicating the consequence of the agent's action.

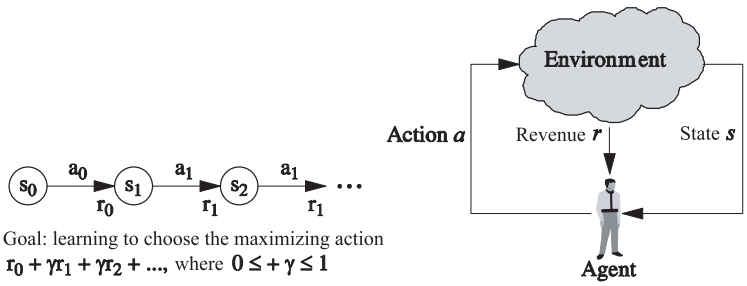

Figure 11.1. *Agent-environment interaction*

In a more formal way, we can designate s ($\in$ S, a finite group) as a representation of the actual environment state, a ($\in$ A, a finite group) as the action chosen and r ($\in$ R, a finite group) as the reward received. The interaction between agent and environment continuously consists of the following sequences:

– the agent observes the current state of environment $s_t \in S$;

– based on state s_t, the agent makes a decision by executing an action $a_t \in A$;

– the environment then makes a transition towards another state $s_{t+1} = s' \in A$ following probability $P_{ss'}(a)$;

– the agent instantly receives a specific revenue $r_t = r(s_t,a_t)$ indicating the consequence of this decision.

The agent's decision process is called policy and it is a function of the group of states to the group of actions (π: S $\rightarrow$ A). The agent must learn a policy π, which makes it possible to choose the next action $a_t = \pi(s_t)$ to execute, based on the current state s_t. The interaction between agent and environment is continuous and the apprentice agent modifies its policy based on its experience and its goal of maximizing the accumulation of rewards in time. This accumulation $V_\pi(s_t)$ achieved by following an arbitrary policy π, from an initial state s_t, is defined as follows:

$$V^{\pi}(s) = E\left\{ \sum_{t=0}^{\infty} \gamma^t r(s_t, \pi(s_t)) \middle| s_0 = s \right\} \qquad [11.1]$$

where E designates operator hope and factor $0 \leq \gamma \leq 1$ represents the temporal propagation constant.

For the agent, the objective is to maximize this sum of received reinforcements and its learning is done by several experiments. The agent is guided in this by different algorithms presented below.

Comment [JL2]: Correct?

11.2.3.1. *Resolution methods*

There are three fundamental method classifications for problem resolution of reinforcement learning: dynamic programming (DP), Monte Carlo (MC) methods and learning by temporal differences (TD) [COR]. Each class has its advantages and drawbacks. DP has well known/studied mathematical foundations but requires a complete and precise environment model. MC methods do not require models and are conceptually simple, but not adapted to an incremental step-by-step calculation. Finally, the TD approach combines the first two methods and uses in this way the

best part of each one. This approach does not require a model and is incremental. These methods also distinguish themselves in terms of convergence speed and efficiency. Below we describe two methods of learning resolution by temporal differences: Q-learning and Sarsa.

11.2.3.1.1. *Q-learning algorithm*

Developed in 1989 by Watkins [WAT 92, WAT 89], Q-learning is part of reinforcement learning methods without model since the object is to learn by experimenting actions to accomplish according to the current state. The agent's goal is to learn a π: S $\rightarrow$ A policy which makes it possible to choose the next action $a_t = \pi(s_t)$ to accomplish based on current state s_t.

For each policy π, we associate a value $Q^{\pi}(s, a)$, that we will call its Q-value and represents the average expected gains if the action has been executed when the current system state is s and that π is then adopted as decision policy. Optimal policy $\pi^*(s)$ is the policy maximizing the accumulation of revenues $r_t = r(s_t, a_t)$ received after an infinite time. Q-learning algorithm's goal is to search for an approximation for $Q^*(s, a) = Q^{\pi^*}(s,a)$ in a recursive manner with only quadruple (s_t, a_t, s'_t, r_t) as available information. This information contains the state at moment t (s_t), state at moment t+1 ($s'_t = s_{t+1}$), the action taken when the system is at state s_t (a_t) and the revenue received at moment t (r_t) following the execution of this action.

Q-values are updated recursively at each transition by using the following formula [11.2]:

$$Q_{t+1}(s,a) = \begin{cases} Q_t(s,a) + \alpha_t \Delta Q_t(s,a), & \text{if } s = s_t \text{ and } a = a_t \\ Q_t(s,a), & \text{otherwise} \end{cases} \qquad [11.2]$$

where:

$$\Delta Q_t(s,a) = \left\{ r_t + \gamma \max_b \left[Q_t(s'_t, b) \right] \right\} - Q_t(s,a) \qquad [11.3]$$

from where we obtain the algorithm summed up in Figure 11.2.

```
Initialize Q0(s, a) to random values
Choose a starting point s0
as long as the policy is not good enough
choose at according to values Qt(st,.)
at = f(Qt(st,.))
obtain in return: st+1 (s') and rt
update Qt+1(st, at) by using formula [11.2]
End as long as
```

Figure 11.2. *Q-learning algorithm*

It is then necessary to set the ratio α in order to gradually establish the policy learned. Factor γ modulates the importance of the rewards expected. In [WAT 92], the authors demonstrate that if each pair (s,a) is infinitely visited and learning rate α leans toward zero, $Q_t(s,a)$ converges to $Q^*(s,a)$ with a probability of 1 when $t \to \infty$. The best policy will then be that with the highest Q-value:

$$\pi^*(s) = \arg \max_{a \in A(s)} Q^*(s,a)$$

The action choice (function f) is not described. It is possible to imagine different selection scenarios (exploration), for example, the action least used, or the one which returns the highest Q-value. In our studies, we have tested a group of exploration methods, such a $\in$-directed method, the Boltzmann method and even random strategies that we will discuss in more detail later.

11.2.3.1.2. *Sarsa algorithm*

Sarsa [SUT 98] is another algorithm for the resolution of a reinforcement learning problem. Contrary to Q-learning, during Q-values update the policy used for choosing the action at moment t is similar to that used for choosing action b at moment t + 1, or:

$$\Delta Q_t(s,a) = \{ r_t + \gamma Q_t(s'_t,b) \} - Q_t(s,a)$$

The exploration policy used can be $\in$-directed, for example. If each pair (s,*a*) is infinitely visited, Sarsa converges to the best policy.

11.2.3.2. *Application of reinforcement learning techniques – state of the art*

Below, we will list a few studies relative to traditional applications (robotics, games, etc.) and to some network applications (routing, CAC, etc.) using

reinforcement learning as their solution. Unfortunately, there are not many studies relative to telecommunications network applications.

The first studies were achieved in the context of learning two-player games (checkers, backgammon, etc.). The other main reinforcement learning field is autonomous robotics, which is a scarcely addressed field in supervised learning because of the difficulty of modeling the real world with enough precision to account for the heterogenity of sensors, ambient noise and the robot-external world dynamic. In [SRI 00], the authors propose a multi-agent system (MAS) able to make economic decisions such as set prices in a competitive market context. They demonstrate that with Q-learning, the system is able to find the best price policy and to lessen the "price war" phenomena between different suppliers.

Studies [BOY 94, MAR 00, MAR 98] are routing propositions in telecommunications networks by using reinforcement learning techniques as a solution. The authors in [BOY 94] propose an extension of Bellman-Ford distance vector algorithm (DBF) which they have called Q-routing. The reinforcement learning module is integrated into each node of a switched network. The routing policy attempts to find the best adjacent node to reach its destination with minimum "transmission time".

Comment [JL3]: Is this abbreviation still correct in the English book or does it need to be changed?

In [SEN 03c], we have proposed an ad hoc routing protocol Q-AOMDV based on one of the most important current ad hoc routing protocols, i.e. AODV (*ad hoc on demand distance vector*) [MANET]. The objective with Q-AOMDV is to balance energy consumption throughout the network by transmitting data traffic in different routes. The choice for the best route is achieved by using an adaptation of the Q-routing algorithm. Let us recall that AODV is an on demand ad hoc routing protocol, which only maintains one route towards destination. Contrary to standard AODV, results show that Q-AOMDV balances energy consumption over the whole ad hoc network while avoiding partitioning of the network into separate sub-networks.

A particularly interesting proposition for resource allocation has been proposed by Nie and Haykin [NIE 99]. We especially focus on this proposition because it is the subject of an example addressed in section 11.4. The authors propose to solve the dynamic resource allocation problem in a GSM type cellular network (voice service only) by using Q-learning.

Comment [JL4]: Correct?

Finally, CAC problems in fixed networks and using reinforcement learning have been addressed in a few studies [MAR 00, MAR 98, MAR 97, RAM 96, MIT 98]. The authors of [MAR 00], for example, propose to solve this problem in a services integration network such as ATM (*asynchronous transfer mode*). The authors put

themselves in the context where a service provider wants to sell network resources to maximize his revenues.

11.3. Call admission control

This section presents an approach which enables the resolution of the call admission control problem in FCA cellular networks supporting several traffic classes.

We recall that a geographical zone served by a mobile network is divided into cells sharing a frequency band in two methods: FCA and DRA [KAT 96]. In FCA, a fixed group of channels is allocated to each cell and a channel is allocated to each user. We also recall that one of the major concerns during mobile network design is the decrease of break probability during intercellular transfer.

Guard channel techniques decrease the probability of communication break by reserving channels for the exclusive use of handoffs in each cell [ZHA 89, KAT 96]. If these techniques are easy to size when we consider one traffic class (one telephone call), they become much more complicated to optimize and are less optimal in a multiclass traffic context.

In fact, in a multiclass traffic context, it is sometimes preferable to block a low priority class call and accept another call belonging to a higher priority class. Call admission control presented in this section enables this type of mechanism. It is obtained by using the reinforcement learning algorithm Q-learning which was presented in the previous section.

11.3.1. *Problem formulation*

We will focus on a simple FCA cell with N available channels and two[2] traffic classes C1 and C2. First class calls require one channel only, whereas second class calls require two channels. This cellular system can be considered as a discrete event system. The main events which can occur in a cell are incoming and outgoing calls. These events are modeled by stochastic variables with appropriate distributions. In particular, new incoming calls and handoffs are Poisson-ruled. Time spent on each call is exponentially distributed. Calls arrive in the cell and leave, either by executing a handoff to another cell, or by terminating normally. The network will then have to choose whether to accept or reject these connection requests. In return,

2 The idea can easily be extended to several traffic classes.

it retrieves revenues received from accepted clients (gains), as well as all revenues received from rejected clients (losses). The objective of the network administrator is to find a CAC policy which will maximize long-term gains and reduce handoff blocking probabilities and losses.

We have chosen the description of all states as $s = ((x_1, x_2), e)$, where x_i is the number of current class C_i calls and e represents a new incoming call or a handoff request in the cell. When an event occurs, the agent must choose one of the possible actions A(s) = {reject, accept}. At the end of a call, no measure need be taken.

For each type of call, revenue is associated with it. Since the main goal of the network administrator is to decrease handoff blocking probabilities, relatively high revenue values are assigned to them. Revenue values for class C1 calls are higher than those for class C2 calls, since C1 is presumed higher priority than C2.

The agent must determine a policy for call acceptance with the only knowledge being the state of the network. This system constitutes an SMDP with a finite set $S = \{s = (x, e)\}$ as space of states and a finite set $A = \{0,1\}$ as space of possible actions where Q-learning is the ideal solution.

11.3.2. *Implementation of algorithm*

After the call admission control problem formulation in the form of an SMDP, we will now describe two Q-learning algorithm implementations able to solve the problem. We have named them TRL-CAC (*table reinforcement learning CAC*) and NRL-CAC (*neural network reinforcement learning CAC*). TRL-CAC uses a simple table to represent function *Q* (the set of Q-values). On the other hand, NRL-CAC uses a network of multilayer neurons [MIT 97]. The differences between these two algorithms (TRL-CAC and NRL-CAC) are explained in terms of memory size and calculation complexity.

Since the approach uses a table, TRL-CAC is the simplest and most efficient method. This approach leads to an exact calculation and it is fully compliant with structures assumptions achieved in order to prove Q-learning algorithm convergence. However, when the group of state-action pairs (s, a) is large or when incoming variables (x, e) constituting state s are continuous variables, the use of a simple table becomes unacceptable because of the huge storage requirements. In this case, some approximation functions, such as state aggregation, neuron networks [MIT 97, MCC 43] or even regression trees [BRE 84] can be used efficiently. The neuron network used in NRL-CAC is made up of 4 entries, 10 hidden units and one output unit.

11.3.2.1. *Implementation*

When a call arrives (new call or handoff), the algorithm determines if quality of service is not violated by accepting this call (by simply verifying if there are enough channels available in the cell). If this quality of service is violated, the call is rejected; if not the action is chosen according to the formula:

$$a = \arg \max_{a \in A(s)} Q^*(s,a) \qquad [11.4]$$

where A(s) = {1 = accept, 0 = reject}.

In particular, [11.4] implies the following procedures: when a call arrives, acceptance Q-value and call reject Q-value are determined from the table (TRL-CAC), or from the neuron network (NRL-CAC). If the reject has a higher value, the call is then rejected. Otherwise, the call is accepted.

Comment [JL5]: Rejection?

In these two cases, and to find out optimal values Q*(s,a), the function is updated at each system transition of state s to state s'. For both algorithms, this is done in the following manner:

– TRL-CAC: [11.2] is the formula used to update the appropriate Q-value in the table;

– NRL-CAC: when a network of neurons is used to store function *Q*, a second learning procedure is required to find out neuron network weights. This procedure uses back propagation (BP) algorithm [MIT 97]. In this case, ΔQ defined by formula [11.3] is used as error signal which is back propagated in the different neuron network layers.

Comment [JL6]: Is this OK or should it be "propagated back"?

11.3.2.2. *Exploration*

For a correct and efficient execution of the Q-learning algorithm, all potentially significant pairs of state-action (s, a) must be explored. For this, during a long enough learning period, the action is not chosen from formula [11.4], but from the following formula [11.5] with a probability of exploration ∈:

$$a = \arg \min_{a \in A(s)} visits(s,a) \qquad [11.5]$$

where visits(s, a) indicate the number of times (s, a) a configuration has been visited. This heuristic, called ∈-directed, significantly accelerates Q-value convergence.

These values are first calculated using this heuristic during a learning period. These values will then be used to initialize Q-values in both CAC algorithms.

11.3.3. *Experimental results*

In order to evaluate the advantages of our algorithms, we have used a discrete event simulation to represent the cellular network. Cell FCA has N = 24 channels. The temporal propagation constant γ has been set to 0.5 and exploration probability $\in$ to 1. Performance criteria used to compare these algorithms are: (i) gains, (ii) losses and (iii) break probabilities. Gains represent the sum of revenues from the acceptance of new calls or handoffs for both traffic classes in the cell. As for losses, they represent the sum of revenues from the rejection of new calls or cellular transfer failure. Break probabilities for all calls taken together (C1 and C2) have been calculated by using the following formula:

$$P_{HO} = \frac{\text{number of system handoff failures}}{\text{number of system handoff attempts}} \quad [11.6]$$

As for break probabilities of traffic classes, they have been calculated for each traffic class C_i by using the following formula:

$$P_{HO(C_i)} = \frac{\text{number of system handoff failures for type } C_i}{\text{number of system handoff attempts for type } C_i} \quad [11.7]$$

We have compared our policies to the one we called greedy policy [TON 00] (policy which accepts a new call or a handoff call if capacity constraint is not violated by accepting this call). We have also compared them to guard channel mechanism. The guard channel mechanism enables the sharing of cell capacity between new calls and handoff calls by giving handoff calls higher priority. This is done by reserving, in each cell, channels for the exclusive use of handoffs (guard channels). The number of reserved channels for handoff calls is a very important parameter in this type of mechanism. In a multiservice context, the question is: how many channels can be reserved for handoffs for each of the traffic classes in order to maximize revenues? To answer this question, we have developed a traditional mathematical model where a cell is represented by a traditional multiserver M/M/N/N-type queue where each server represents a communication channel.

We have carried out a set of simulations including: (i) constant traffic load for all traffic classes, (ii) variable traffic classes and (iii) variable traffic load in time. We will not present all the results obtained, but for more information see [SEN 03a, SEN 03b]:

– constant traffic load: the first experiment considers a constant traffic load for both C1 and C2 classes. The simulation parameters used are the same as those used during the learning period;

– variable traffic load: in this second experiment we have used the same policies as the previous experiment (constant traffic load), but with six different traffic loads (for both C1 and C2 classes);

– variable traffic load in time: in this last experiment we have again used the same policies as the first experiment, but with a variable traffic load in time. Indeed, traffic load in a cellular system is variable in one day. We have used the traffic model presented in Figure 11.3 describing the variation of incoming rates during a working day. Rush hours appear at 11am and 4pm.

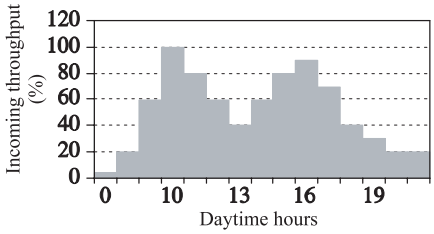

Figure 11.3. *Traffic model for a working day*

For all these experiments, the results have confirmed that the algorithms proposed are more efficient than the other heuristics for all traffic loads, in particular when traffic load is high. For example, Figure 11.3 shows simulation results with the assumption that both traffic classes used the same traffic model. Call blocking probabilities were calculated on an hourly basis. We have compared our algorithms to: guard channel with fixed thresholds (these thresholds have been calculated for a given constant traffic load) and guard channel with optimized thresholds (these thresholds have been calculated for each traffic load value). Improvements to proposed algorithms are apparent compared to greedy policy, particularly during traffic peaks (around 11am and 4pm). We notice that guard channel with optimized thresholds gives better results than the two algorithms based on Q-learning. On the other hand, this method presumes that optimized thresholds are calculated offline for

each traffic load value. On the contrary, admission control algorithms based on Q-learning (TRL-CAC and NRL-CAC) have adaptation and generalization capabilities, so optimal Q-value values are not recalculated for each traffic load.

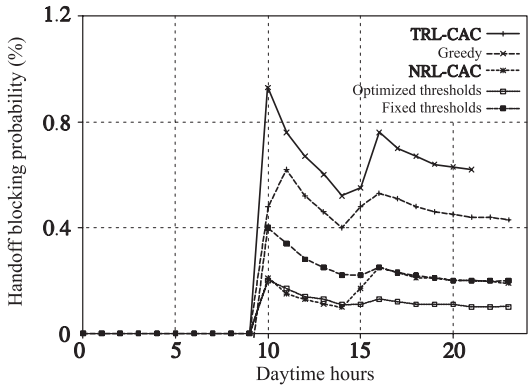

Figure 11.4. *Probability of handoff blocking with a variable traffic load in time*

11.4. Dynamic resource allocation

In this section, we will present a new mechanism enabling the resolution of dynamic resource allocation, also taking into account call admission control in DCA systems supporting several traffic classes.

Comment [MLM7]: Should this be DRA?

Let us recall that in dynamic resource allocation (DRA) strategies, all available channels in the system are put in a common pool which can be used by all base stations [KAT 96]. During a communication request, a cell chooses a common pool[3] channel, which will be restored at the end of the call.

Different dynamic resource allocation algorithms have been compared in terms of performance, flexibility and complexity. One of the most widely used strategies in other works is a class of algorithms called “exhaustive searching DRA” [NIE 99, ZHA 89, COX 72, DEL 93, DIM 93, SIV 90]. In this type of algorithm, a reward (cost) is associated with each available channel. When a new call arrives, the system exhaustively searches for a channel with the highest reward (lowest cost) and assigns it to the call. Most of these propositions do not consider CAC policies as a method of preventing congestion [LEV 97]. The relation between resource

3 While respecting signal/interference relation *C/I*.

allocation and admission control has been previously studied [NAG 95, TAJ 88, YAN 94].

This section presents a new approach to solve the dynamic resource allocation problem, also considering call admission control in DRA systems. For reasons mentioned in the previous section, call admission control is vital when the network supports several classes of clients. Allocation policies are obtained by using the reinforcement learning algorithm Q-learning. The main functions of the proposed mechanism called Q-DRA presented in this section are: accepting clients and rejecting others, and allocating the best available channel for accepted clients. The goal is to maximize accumulation of received revenue through time.

We will now briefly describe the problem formulation by a SMDP and the implementation of the Q-learning algorithm capable of resolving this SMDP.

11.4.1. *Problem formulation*

This contribution is an extension of the study by Nie and Haykin [NIE 99] and is part of exhaustive searching DRA algorithms. We are considering the resolution of the dynamic resource allocation problem as well as the call admission control problem in a cellular network. This network contains N cells and M channels available which are maintained in a common pool. It supports two traffic classes (contrary to [NIE 99] where only one traffic class – telephone call – was taken into account). Each channel can be temporarily allocated to any cell, as long as the constraint on reuse distance is satisfied (a given signal quality must be maintained).

The calls arrive in the cell and leave based on appropriate distributions. The network will then choose to accept or reject these connection requests. If the call is accepted, the system allocates one of the available channels to it from the common pool. The goal of the network administrator is to find a dynamic resource allocation policy capable of maximizing long-term gains and decrease probabilities of handoff blocking (contrary to [NIE 99] which does not give any priority to handoff calls).

We have chosen the group of states as being $\{s = (i, D(i), (x_1, x_2), e_i)\}$, where D(i) represents the number of available channels in cell i where event e_i has occurred, x_k represents the number of current calls from class C_k and e_i indicates the arrival of a new call or a handoff call in a cell i. When an event occurs, the agent must choose one of the possible actions $A(s) = \{0 = \text{reject}\} \cup \{1,\ldots, M\}$. When a call ends, no measure has to be taken. The agent will have to determine the policies for accepting or rejecting a call and, when accepted, to allocate the channel that will enable the maximization of gain accumulation received by only knowing the current

network state s. This system constitutes an SMDP which has as space of states a finite group S = {(i, D(i), x, e)} and as space of possible actions a finite group A = {0, 1…, M}. The choice of revenues that we have used considers intercellular interferences and its consequence is a situation in which channels already used in compact cells[4] [ZHA 91] will have a better chance of being chosen. Contrary to other studies, Q-DRA considers the call type and grants priority to handoff calls.

11.4.2. *Algorithm implementation*

After the problem formulation in the form of SMDP, we will describe implementation of the Q-learning algorithm capable of its resolution. The cellular system studied is made up of N = 36 hexagonal cells and M = 70 channels available in a common pool. We use a reuse distance $D = \sqrt{21}R$ (R represents cell radius). This implies that if a channel is allocated to a cell i, then it cannot be reused in the two rows adjacent to i because of co-channel interferences. In this way, there are at most 18 cells interfering with each system cell. The temporal propagation constant γ has been set to 0.5 and the learning rate α to 0.1.

In the previous section, we have used a network of neurons to represent Q-values (NRL-CAC), but this time we have chosen an approximation based on state aggregation. Instead of precisely defining the number of calls x_i for each traffic class C_i, we have chosen to characterize traffic as follows: low, medium, high. The space of states is thus reduced and a simple table can be used to represent aggregated states. Because identical states (or states with a similar number of current calls) have the same Q-values, performance loss linked to aggregation becomes insignificant [TON 99].

11.4.2.1. *Implementation*

When a call arrives (new call or handoff call) in cell i, the algorithm determines if quality of service is not violated by accepting this call (by simply verifying if there are free channels in the common pool). If this quality of service is violated, the call is rejected, otherwise the action is chosen depending on the following expression:

$$a = \arg \max_{a \in A(s)} Q^*(s, a) \qquad [11.8]$$

4 Compact cells are cells with an average minimum distance between co-channel cells.

where A(s) = {0 = reject, 1, 2…, M}.

Formula [11.8] implies the following procedures. When there is a call connection attempt in cell i, Q-value of reject (a = 0) as well as acceptance Q-values (a = 1, 2…, M) are determined from the Q-value table. Acceptance Q-values include the different Q-values corresponding to choices of each channel a (a = 1, 2…, M) to serve the call. If the rejection has the highest value, then the call is rejected. Otherwise, if one of the acceptance values has the highest value, the call is accepted and channel a is allocated to it.

11.4.2.2. *Exploration*

For a correct and efficient execution of Q-DRA algorithm, the action is not chosen from formula [11.4], but based on a Boltzmann distribution [WAT 89] during a relatively long learning period. The idea is first to favor exploration (the probability of executing actions other than those with the highest Q-value) by using all possible actions with the same probability. Then the goal is to gradually move towards the use of the action with the highest Q-value. The values learned will be used later to initialize Q-values in Q-DRA.

11.4.3. *Experimental results*

In order to study Q-DRA performances, a group of simulations was completed. We have compared Q-DRA to greedy-DRA[5] policy [TON 99], as well as to the DRA-Nie algorithm [NIE 99]. Algorithm performances have also been evaluated in terms of gains, losses, as well as handoff blocking probabilities. A group of simulations was carried out including: a case of traffic load evenly distributed over all cells, a case of load not evenly distributed, a case of variable traffic load in time and a case of equipment failure. We will not present all the results obtained (see [SEN 03a, SEN 03c] for more information):

– even traffic distribution: the first experiment considers a constant traffic load in the 36 cells for both traffic classes. We have used policies learned during the learning period, but with five different traffic loads (for both C1 and C2 classes);

– uneven traffic distribution: in this second experiment, we have used policies learned during the learning period but traffic loads in this case are no longer evenly distributed over the 36 cells. The average traffic load considered was of 7.5 Erlangs;

5 Greedy-DAC: policy which randomly chooses a channel to serve a call with no measure of interference. Each M channel has the same probability to be chosen for serving the new call.

– variable traffic load in time: in the third experiment, we wanted to test Q-DRA performances when traffic load changes in time;

– equipment failure in a DRA system: in the last experiment, we have simulated equipment failure due to some channels becoming temporarily unavailable. At the beginning of the simulation there are 70 available channels in the system. However, between 10am and 3pm, we have temporarily suspended 0, 3, 5 and 7 channels.

For all these experiments, results show the possibilities that reinforcement learning offers in order to learn the best admission and dynamic resource allocation policy. Policy results using Q-DRA indicate significant improvements compared to other policies. These improvements are also slightly better than those from DRA-Nie. Q-DRA has an aptitude for generalizing and adapting to changes in traffic conditions. For example, Figure 11.9 shows the impact of channel failure over call break probabilities by using the Q-DRA algorithm. We notice that Q-DRA has a certain robustness against equipment failure situations and easily adapts, especially in the case where 3/5 channels have been suspended.

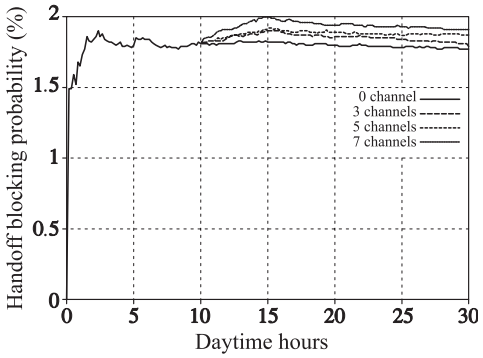

Figure 11.5. *Q-DRA performance during channel failure*

11.5. Conclusion

The first cellular networks were mainly designed to offer telephone service. Current cellular systems promise a diversification of services offered with clearly superior throughput. Service diversification (voice, SMS, multimedia services, Internet access, etc.) requires several levels of quality of service (QoS) to guarantee. However, availability of radio resources during a call is not necessarily guaranteed and mobile users can also experience service degradation/break. Since break/degradation of a current call belonging to a high priority service class is

generally less desirable than break/failure of a call connection belonging to a lower priority class, new mechanisms of CAC are vital. In fact, efficient call admission control is required to prevent this limitation of available radio resources in the cellular network radio interface. Efficient management of radio resources through dynamic allocation policies also proves to be essential to prevent this type of problem. These new mechanisms consist of defining resource management rules for each traffic class, for the optimization of usage rate and to satisfy the multiple QoS constraints.

In order to prevent this type of problem, several propositions exist in other works and their main goal is to avoid inconveniences caused by communication breaks for users. However, we have noticed that these solutions often ignore experience and knowledge which could be acquired during real-time system execution.

The contributions presented in this chapter, relative to cellular networks, are intended to benefit from this experience and knowledge in order to optimize problems encountered in cellular networks. In the first contribution we needed to find a new approach to solve the CAC problem in a multiservice cellular network where channels are permanently allocated to cells. For the second, we needed a new approach to the dynamic resource allocation problem in a multiservice cellular network. This last contribution is ingenious, since it combines optimal CAC policy research and the best dynamic channel allocation strategy. These proposed mechanisms to solve such complex problems as those linked to cellular networks use reinforcement learning as their solution. These are creative and intelligent solutions. In addition to the creativity of these mechanisms, the advantages gained by using such approaches can be summarized as follows. Contrary to other studies (studies based on mathematical models or simulations, presuming fixed experimental parameters), these solutions are adaptable to variations of network state (i.e. variations of traffic conditions, equipment failure, etc.). Because of their distributed nature, they can easily be implemented in each base station, which makes them more attractive. Channel admission control and dynamic allocation tasks are quickly determined with little calculation efforts. They are obtained by a simple specification of preferences between service classes. We have also demonstrated, with a large group of experiments, that these mechanisms give the best results compared to other heuristics. These are distributed algorithms and signaling information exchanged between base stations are almost null. These mechanisms are therefore more attractive because of their implementation simplicity.

Finally, we can say that our main contribution has been to propose and test mechanisms for solving problems encountered in mobile networks (CAC and dynamic resource allocation). We have been able to demonstrate that it is possible to

use techniques from AI and, more specifically, learning techniques in order to develop efficient, robust and easy to implement mechanisms.

11.6. Bibliography

[AGH 01] AL AGHA K., PUJOLLE G., VIVIER G., *Réseaux de mobiles et réseaux sans fil*, Eyrolles Publishing, 2001.

[BOY 94] BOYAN J.A., LITTMAN M.L., "Packet routing in dynamically changing networks: a reinforcement approach", *Advances in Neural Information Processing Systems (NIPS'94)*, vol. 6, p. 671-678, San Mateo, 1994.

[BRE 84] BREIMAN L., FRIEDMAN J.H., OLSEN R.A., STONE C.J., *Classification and Regression Trees*, Chapman & Hall, 1984.

[CAL 92] CALHOUN G., *Radio cellulaire numérique*, Tec & Doc, 1992.

[COR] CORSINI M.M., Class on reinforcement learning, available at: http://www.sm.u-bordeaux2.fr/~corsini/Cours/HeVeA/rl.html.

[COX 72] COX D.C., REUDINK D.O., "Dynamic channel assignment in two dimensional large mobile radio systems", *Bell Syst. Tech. J.*, vol. 51, p. 1611-1627, 1972.

[DEL 93] DEL RE E., FANTACCI R., RONGA L., "A dynamic channel allocation technique based on Hopfield neural networks", *IEEE Trans. Vehicular Technology*, vol. 45, p. 26-32, February 1996.

[DIM 93] DIMITRIJEVIC D.D., VUCETIC J., "Design and performance analysis of the algorithms for channel allocation in cellular networks", *IEEE Trans. Vehicular Technology*, vol. 42, p. 526-534, November 1993.

[GIB 96] GIBSON J.D., *The Telecommunications Handbook*, IEEE Press, 1996.

[KAT 96] KATZELA I., NAGHSHINEH M., "Channel assignment schemes for cellular mobile telecommunications systems", *IEEE Personal Communications Magazine*, June 1996.

[LEV 97] LEVINE D.A., AKYILDIZ I.F., NAGHSHINEH M., "A Resource Estimation and Call Adaptation Algorithm for Wireless Multimedia Networks Using the Shadow Cluster Concept", *IEEE/ACM Transactions on Networking*, vol. 5, no. 1, p. 1-12, February 1997.

[MANET] IETF MANET Working Group (*mobile ad hoc networks*), www.ietf.ora/html.charters/manet-charter.html.

[MAR 97] MARBACH P., TSITSIKLIS J.N., "A Neuro-Dynamic Approach to Admission Control in ATM Networks: The Single Link Case", *ICASSP'97*, 1997.

[MAR 98] MARBACH P., MIHATSCH O., SCHULTE M., TSITSIKLIS J.N., "Reinforcement learning for call admission control and routing in integrated service networks", in JORDAN M. *et al.*, (ed.), *Advances in NIPS 10*, MIT Press, 1998.

[MAR 00] MARBACH P., MIHATSCH O., TSITSIKILS J.N., "Call admission control and routing in integrated services networks using neuro-dynamic programming", *IEEE Journal on Selected Areas in Communications (JSAC'2000)*, vol. 18, no. 2, p. 197-208, February 2000.

[MCC 43] MCCULLOCH W.S., PITTS W., "A logical calculus of the ideas imminent in nervous activity", *Bulletin of Math. Biophysics*, vol. 5, 1943.

[MIT 97] MITCHELL T.M., *Machine Learning*, McGraw-Hill, 1997.

[MIT 98] MITRA M.I., REIMAN J., WANG, "Robust dynamic admission control for unified cell and call QoS in statistical multiplexers", *IEEE Journal on Selected Areas in Communications (JSAC'1998)*, vol. 16, no. 5, p. 692-707, June 1998.

Comment [JL8]: Author's initials are missing – please add

[NAG 95] NAGHSHINEH M., SCHWARTZ O., "Distributed call admission control in mobile/wireless networks", *PIMRS, Proceedings of Personal Indoor and mobile radio communications*, 1995.

[NIE 99] NIE J., HAYKIN S., "A Q-Learning based dynamic channel assignment technique for mobile communication systems", *IEEE Transactions on Vehicular Technology*, vol. 48, no 5, September 1999.

[RAM 96] RAMJEE R., NAGARAJAN R., TOWSLEY D., "On Optimal Call Admission Control in Cellular Networks", *IEEE INFOCOM*, p. 43-50, San Francisco, March 1996.

[SEN 03a] S. SENOUCI, Application de techniques d'apprentissage dans les réseaux mobiles, PhD Thesis Pierre and Marie Curie University, Paris, October 2003.

[SEN 03b] SENOUCI S., BEYLOT A.-L., PUJOLLE G., "Call Admission Control in Cellular Networks: A Reinforcement Learning Solution", *ACM/Wiley International Journal of Network Management*, vol. 14, no. 2, March-April 2003.

[SEN 03c] SENOUCI S., PUJOLLE G., "New Channel Assignments in Cellular Networks: A reinforcement Learning Solution", *Asian Journal of Information Technology (AJIT'2003)*, p. 135-149, vol. 2, no. 3, Grace Publications Network, July-September 2003.

[SIV 90] SIVARAJAN K.N., MCELIECE R.J., KETCHUM J.W., "Dynamic channel assignment in cellular radio", *Proc. IEEE 40th Vehicular Technology Conf.*, p. 631-637, May 1990.

[SRI 00] SRIDHARAN M., TESAURO G., "Multi-agent Q-learning and Regression Trees for Automated Pricing Decisions", *Proceedings of the Seventeenth International Conference on Machine Learning (ICML'00)*, Stanford, June-July, 2000.

[SUT 98] SUTTON R.S., BARTO G., ANDREW, *Reinforcement Learning: An Introduction*, MIT Press, 1998.

[TAJ 88] TAJIMA J., IMAMURA K., "A strategy for exible channel assignment in mobile communication systems", *IEEE Transaction on Vehicular Technology*, vol. 37, p. 92-103, May 1988.

[TON 99] TONG H., Adaptive Admission Control for Broadband Communications, PhD Thesis, University of Colorado, Boulder, Summer 1999.

[TON 00] TONG H., BROWN T.X., "Adaptive Call Admission Control under quality of service Constraint: a Reinforcement Learning Solution", *IEEE Journal on Selected Areas in Communications (JSAC'2000)*, vol. 18, no. 2, p. 209-221, February 2000.

[WAT 89] WATKINS C.J.C.H., Learning from delayed rewards, PhD Thesis, University of Cambridge, Psychology Department, 1989.

[WAT 92] WATKINS C.J.C.H., DAYAN P., "Q-learning", *Machine Learning*, vol. 8, p. 279-292, 1992.

[YAN 94] YANG W.B., GERANIOTIS E., "Admission policies for integrated voice and data traffic in CDMA packet radio networks", *IEEE Journal on Selected Areas in Communications*, vol. 12, p. 654-664, May 1994.

[ZHA 89] ZHANG M., YUM T.S., "Comparisons of channel assignment strategies in cellular mobile systems", *IEEE Trans. Vehicular Technology*, vol. 38, no. 1, p. 211-215, June 1989.

[ZHA 91] ZHANG M., YUM T.S., "The non-uniform compact pattern allocation algorithm for cellular mobile systems", *IEEE Trans. Vehicular Technology*, vol. 40, no. 2, p. 387-391, May 1991.