

# Dynamic Decision Making for Candidate Access Point Selection

Burak Simsek<sup>1</sup>, Katinka Wolter<sup>1</sup> \*, and Hakan Coskun<sup>2</sup>

<sup>1</sup> Institut für Informatik, HU Berlin  
Unter den Linden 6, 10099 Berlin

simsek@informatik.hu-berlin.de, wolter@informatik.hu-berlin.de

<sup>2</sup> ETS, TU Berlin Franklinstr.28/29  
10587 Berlin

coskun@cs.tu-berlin.de

**Abstract.** In this paper, we solve the problem of candidate access point selection in 802.11 networks, when there is more than one access point available to a station. We use the QBSS (quality of service enabled basic service set) Load Element of the new WLAN standard 802.11e as prior information and deploy a decision making algorithm based on reinforcement learning. We show that using reinforcement learning, wireless devices can reach more efficient decisions compared to static methods of decision making which opens the way to a more autonomic communication environment. We also present how the reinforcement learning algorithm reacts to changing situations enabling self adaptation.

## 1 Introduction

The existence of more than one access point available to one station is usual for many daily applications of WLAN. In such cases, the station should be able to decide on one of the access points optimally. At this point, some kind of information that is helpful for being able to estimate to what extent the candidate access points can satisfy the required service quality is essential. This requires the knowledge of the most informative network parameters among many possible. Currently, the QBSS (quality of service enabled basic service set) Load Element of the WLAN standard 802.11e [8] seems to be the most appropriate information element that could be used to choose the access points which can offer the required QoS (quality of service). It is responsible for informing the stations about the load of an access point so that stations can decide on associating themselves with the access point or not. Including three parameters, this element is periodically sent to all of the stations within the beacon frames. These parameters are station count, available admission capacity and channel utilization.

However, in our previous study we showed by correlation analysis that the parameters of the QBSS load element are not reliable in most of the cases [14].

---

\* The authors Burak Simsek and Katinka Wolter would like to thank German Research Foundation (DFG) for support of this work under grant number WO 898/1-2.

This is due to the fact that there is no high correlation with the QBSS load element parameters and the QoS metrics such as delay, jitter and loss. Unfortunately, choosing an access point with less channel utilization or less stations associated with the access point is the most used approach which does not lead to efficient decision making in 802.11e networks.

In this paper we find out the applicability, reliability and performance of a more sophisticated decision making algorithm for the candidate access point selection problem which is based on reinforcement learning. We show that the reinforcement learning algorithm (RLA) enables the stations to reach higher decision accuracy compared with the traditional methods. We deploy regression analysis for performance comparison purposes since regression analysis includes but not restricted to the traditional logic such as "choose the access point with less station count or channel utilization". Unfortunately there is no other study known to the authors which deals with the use of intelligent decision making methods for the candidate access point selection problem. We demonstrate that the algorithm does not pose additional load to the system during runtime if it is trained prior to the deployment by the end customers of wireless devices implementing 802.11e. Following this, we propose an approach to engage RLA over wireless devices in an efficient manner. We show how a trained algorithm can adapt itself to dynamic environments which introduces a significant advantage when compared with the statistical way of decision making and therefor enables an autonomic communication environment.

Since the 802.11e standard is new, most of the relevant studies made so far include performance analysis and improvement [3]. To the best of our knowledge, there is no study working on the candidate access point selection problem and the evaluation of the QBSS load element in solving this problem. Additionally, although there are services enabling intelligent decision making over 802.11e networks, there is no study known to us in this direction. Reinforcement learning is a well known algorithm of artificial intelligence which has been used very often in different fields of scientific research. Soccer robots [5] and chess and backgammon tools [6] are some examples. The main reason why we applied reinforcement learning to the candidate access point selection problem is that it is relatively easier to deploy over wireless devices. It is less complex when compared with other learning algorithms of artificial intelligence. [11] gives an enhanced study of such algorithms. However despite their complexities and problems in solving infinite size problems [15], simplified versions of evolutionary algorithms [9] might also prove to be effective which is another research problem. The main merit of this study is that it prepares the basics for a network management methodology by the means of a dynamic decision making algorithm over 802.11e. We also deploy the same algorithm for call admission control using the traffic specification element of the protocol IEEE 802.11e. Due to size restrictions, the results of admission control with reinforcement learning are going to be presented within our next study.

The rest of the paper is structured as follows. In the second section, we introduce the candidate access point selection problem that we want to solve and

summarize the decision making algorithms. In the third section we describe the simulation environment. The results of the simulation are given in the fourth section. In the fifth section we present how RLA can adapt to dynamic environments. The sixth section makes the conclusion by comparing the results and consequently making recommendations.

## 2 Decision Making

### 2.1 Regression Analysis

The main goal of regression analysis is to estimate the effect of a change in one of the independent variables of a system on a dependent variable within the system by using some observations of the dependent and the independent variables. One of the simplest forms of regression analysis is the linear regression as given in equation (1).

$$y_i = a_0 + a_1x_{i,1} + a_2x_{i,2} + \dots + a_nx_{i,n} \quad i < m \quad (1)$$

where  $n$  is the number of independent variables,  $m$  the number of dependent variables,  $y_i$  the dependent variables,  $x_{i,k}$  the independent variables and  $a_k$  the parameters to be estimated. During our study, we tried a number of possible functions like linear, log-linear or non-linear functions in order to estimate the expected QoS using the QBSS load element parameters. In this paper we selected to present only the results of the linear function because despite its simplicity, the amount of correct decisions was the highest in most of the cases. A typical equation for QBSS load elements is given as follows:

$$E[QoS] = a * StaCount + b * ChanUtil + c * AdCap + d \quad (2)$$

where  $StaCount$  is the number of stations associated with an access point,  $ChanUtil$  is the percentage of time that the channel being used by the access point is sensed to be busy,  $AdCap$  is the remaining amount of time in the HCCA scheduler and  $d$  is the disturbance term. We try to estimate  $a, b, c$  and  $d$  for this function and use simulation results as the input data for this purpose. Although the station count and channel utilization appear to have a reverse ratio to the expected QoS and available admission capacity a direct ratio, we showed in [14] that this assumption is not true in many cases for 802.11e. Additionally during regression analysis, the parameters of the variables having a reverse ratio to the dependent variable are negative. Because of these two reasons, it makes sense to have the variables, which we expect to have a reverse ratio with QoS, as a nominator within the equation (2).

### 2.2 Reinforcement Learning

A formal definition of a reinforcement learning problem in our case can be written as follows: An RL decision making algorithm receiving the QBSS load element

information is at state  $s_t(x, y, z \mid x \in \mathbb{Z}, y \in 0..255, z \in 0..65536)$  where  $x$  is the total station count,  $y$  is the normalized channel utilization of the access point and  $z$  is the amount of medium time available in units of 32 microseconds, and has to make an action  $a_t$  (associate, do not associate). Each action  $a_t$  brings reward  $R_t$  to the algorithm. The goal of the RL algorithm is to select actions which maximize the sum of its discounted rewards  $R_t$ , where the discount factor is  $\gamma \in [0, 1]$ . In order to do this, RL algorithm has to decide about the value of an action at each state which is given as  $Q(s_t, a_t)$ . The optimal  $Q(s_t, a_t)$  satisfies the following equation:

$$Q(s_t, a_t) = \left( R_t + \gamma \sum_{j=S} P_{s,j}(a_t) V(j) \right) \quad (3)$$

Here,  $V(j)$  is the maximum  $Q$  value that is expected to occur after a transition from  $s_t$  to  $j$  with respect to all actions  $a_t$ .  $P_{s,j}(a_t)$  is the transition probability from state  $s_t$  to state  $j$ . For further information about our RL algorithm and the choice of the parameters please see [13].

### 2.3 Problem Definition

We define two dynamic decision making problems.

1. Given an access point at state  $s_t$ , the decision function  $f(s_t)$  which might be the function of the regression analysis or the RLA, a desired QoS level  $QoS^*$  and the real QoS after simulations  $QoS$ ,
  - if  $f(s_t) \geq QoS^*$  then decision = accept else decision = not accept.
  - if decision = accept and  $QoS \geq QoS^*$  then decision is accurate else decision is false.

This problem is equivalent to estimating if an access point is going to be able to provide a service with sufficient QoS.

2. Given two access points at states  $s_{1,t}$  and  $s_{2,t}$ 
  - if  $f(s_{1,t}) \geq f(s_{2,t})$  then decision = 1 else decision = 2.
  - if  $QoS_1 \geq QoS_2$  then decision is accurate else decision is false
 which is equivalent to finding the better access point. The problem is not different when there are more access points. In such cases expected QoS for each access point should be calculated independently and then compared with each other.

We use the percentage of accurate decisions made by regression analysis and the RLA in order to decide on:

- to what extent the QBSS load element can help in choosing the right access point,
- which one of the decision making algorithms (reinforcement learning and regression analysis) help making more accurate decisions and in which instances.

Although a decision making method is successful in one of these problems, this might not be the case for the other problem. For example, even though a decision making algorithm can estimate which access point is better than the other, this does not guarantee that it can also estimate the real level of QoS. This is also true vice versa. Hence we have to differentiate between those two problems throughout the paper.

### 3 Simulation

#### 3.1 Simulation environment

The simulated network consists of one or two QoS enabled access points (QAP) depending on which one of the above defined problems is being solved and a random number of stations associated with the QAPs. Each station sends one type of traffic stream from the the list given below where up to 7 voice, 3 video, 10 interactive and 10 background streams are allowed to associate with an access point. Network is simulated using a slightly modified version of the 802.11e ns2 model of Ni [4].

In [12] it was shown that the current public hotspots are mainly occupied with low load TCP flows and only a few of the connected users demand real time applications based on high load UDP traffic. Additionally, the percentage of real time services is going to increase as the number of applications in this area increases [7]. Hence the network environment is designed to include 4 different traffic types. The first type is defined for voice traffic with the codec G.711. The second priority is defined for video traffic with different qualities which comprises most of the video codecs being used in the internet [10]. Third and fourth traffic types are defined to simulate normal hot spot user behavior as given in [12]. These traffic types are defined as follows:

1. Bidirectional constant bit rate (CBR) voice traffic using UDP with a packet size of 160 bytes and packet interval 20ms (8 Kbytes/s) corresponding to the VoIP codec G.711. (1st access category)
2. 12 simulated VBR video traffic streams using UDP with minimum packet size of 28 and maximum packet size of 1024 bytes with an average packet interval of 23ms corresponding to 30Kbytes/s. (2nd access category) (Average Quality Video)
3. Bidirectional interactive traffic using TCP with a packet size of 1100 bytes and exponentially distributed arrival rates having an average of 50ms on time, 30ms off time and sending rate of 60Kbits/s during on times corresponding to an average of 10Kbytes/s. This complies with the interactive traffic definition of 3GPP TS 22.105 and ITU G.1010. (3rd access category)
4. VBR Background traffic using TCP with a packet size of 1200 bytes and exponentially distributed inter arrival times having an average of 1000ms off and 200ms on times with a sending rate of 100Kbits/s corresponding to low load 11Kbytes/s traffic. (4th access category) (3GPP TS 22.105 Web Browsing- HTML definition.)

**Table 1.** List of Simulation Parameters

Bandwidth	11Mbps
PLCPTransmissionrate	1 Mbps
RTSThreshold	3000 $\mu$ s
ShortRetryLimit	7
LongRetryLimit	4
slotTime	9 $\mu$ s
AIFS(1,2,3,4)	1, 2, 6, 12
CWmin(1,2,3,4)	7, 15, 15, 31
CWMax(1,2,3,4)	15, 31, 255, 525

Additionally the 802.11e specific parameters are given in table 1.

The type of the traffic we are interested in throughout the paper is the voice traffic. As a metric of QoS for voice traffic we use the mean opinion score (MOS) values defined in ITU-T Rec. G.107 which is the widely accepted metric of industrial organizations to measure the quality of VoIP applications [2, 1]. MOS rates voice calls on a scale of 1 to 5. It shows the satisfaction of real people who use a connection and rate the quality of voice signal after it has passed through a network from a source (transmitter) to a destination (receiver).

The most dominant enhancement in 802.11e MAC is the introduction of two new functions, the enhanced distributed channel access (EDCA) for differentiated services and the HCF controlled channel access (HCCA) for integrated services. Although HCCA and EDCA are compulsory within the standard, the amount of time reserved for these functions can be varied. In [14] it was shown that the percentage of time used by HCCA has a significant impact on the resultant QoS. For this reason, we also included the ratio of the time reserved for HCCA as a simulation parameter in addition to traffic types defined above. For more detail about the HCCA and EDCA please refer to [8].

### 3.2 Simulation with Regression Analysis

The simulation of decision making in 802.11e networks is composed of two periods. The first period is called the training period and the second one is called the decision period. Training period is used for historical data collection and evaluation while the decision period is the verification of the learned cases.

Training period for regression analysis means the parameter estimation procedure which is done as follows. We start one simulation run with a set of traffic streams selected from the traffic types given above. At the 45<sup>th</sup> simulation second we start measuring the QBSS load element parameters for five seconds. In 802.11e, the duration of time in which QBSS load element measurement is realized is left to vendors. Since 5 seconds proved to be enough for the convergence of the QBSS load element parameters in all cases, we made measurements for the last five seconds. After each simulation run we calculate the MOS values of each voice stream within the run using the measured delay, jitter and loss rates. This gives us the dependent variable of our regression analysis. We also note the station count, channel utilization and available admission capacity during the

last five seconds of each run as our independent variables. After running more than 20000 such simulation runs, we make curve fitting using equation (2) and estimate the parameters a,b,c and d.



**Fig. 1.** One simulation run within the training period of the regression analysis

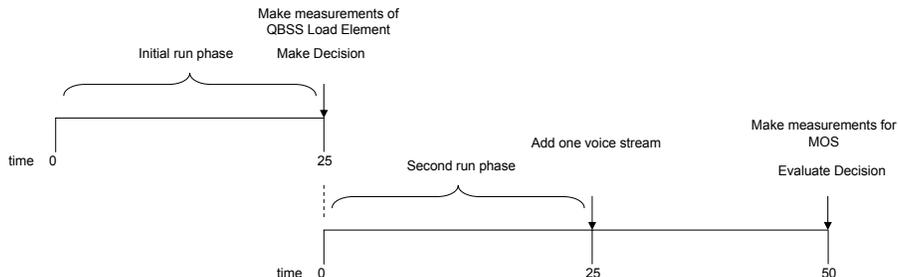
During the decision period we simply use the estimated formula and make decisions for the problems as defined in section 2.3.

### 3.3 Simulation with Reinforcement Learning

A training step for reinforcement learning for the first problem of section 2.3 is illustrated in figure 2. We can summarize this training step as follows: A number of stations with different traffic types are connected to the access point or access points and start transmitting their packets. Initially the simulation is run for 25 seconds. Afterward, the QBSS load element and the number of traffic streams of different priority levels are found. At this point, the decision making algorithm decides on adding a new voice stream to the simulation or not by considering the information it is given (either the QBSS load element or the number of traffic streams associated with different priority levels). The decision is made either using the previously learned actions with a probability of  $(1-\alpha)$  or randomly with a probability of  $\alpha$ . The optimum choice of the parameter  $\alpha$  is done by making cross-validation analysis together with the parameter  $\gamma$  of the equation (3). For all the results presented in this study we used 0.25 for  $\alpha$  and 0.82 for  $\gamma$ , since these values proved to give the best performance. Regardless of the result of the decision, a new simulation with the same configurations is started. At time 25, the new voice stream is joined to the system. After this point, the simulation goes on for 25 seconds and stops at 50 seconds. Jitter, delay, and loss rates are calculated using the last 25 seconds' information and correspondingly MOS values are calculated. If the calculated MOS value is over the threshold which is 3.8 in our case and if the decision of the algorithm was accepting the new traffic, a true decision was given and the algorithm is rewarded directly proportional to the difference between the MOS value and the threshold for MOS. Correspondingly the  $Q(s_t, a_t)$  value of the state and the action decided by the algorithm is modified using equation (3). This is one run during training period of the first question of section 2.3. We train the RLA with 1000 such runs.

In fact the number of runs during the training period affects the accuracy of the decisions after training in case the training data does not fluctuate very much [13]. However if there are fluctuations, it makes sense to use less training data to capture the effects of such fluctuations as it will be mentioned in section 5. In our case the learned Q function values converged to a steady state within less than 1000 runs. However we also present results with 100 runs in section 5.

Additionally learning is also dependent on the reward function which is strictly dependent on the MOS value. Cases having MOS values much lower or higher than the threshold value are learned much better than cases with MOS values near to the threshold value. Such configurations of the RLA are problem specific and we won't go into more detail in this study because of size limitations.



**Fig. 2.** One simulation run within the training period of the RL algorithm

For the training step of the second question we use two simultaneous simulation runs with two access points. In one of these simultaneous runs, decision is made using the RLA and in the other one decision is always the opposite of the decision given by the RLA. In this way, we can compare the results of opposite decisions and hence decide which action would be accurate. At the end of each run we compare the MOS values of both cases. If an accurate decision is given by the RLA, it is rewarded directly proportional to the difference between the MOS values of both cases. The learning matrix of the RLA is modified by changing the expected values of all related actions with respect to this reward. The  $Q(s_t, a_t)$  value of the state and the action decided by the algorithm is modified and a new learning round is started as explained above.

The information learned during the training period, namely  $Q(s_t, a_t)$ , is used in order to make decisions during the decision period, by simply choosing the action with a greater  $Q(s_t, a_t)$  value.

## 4 Simulation Results

### 4.1 Results of Regression Analysis

Table 2 summarizes the regression analysis using equation (2) and the information given by the QBSS load element. In our previous research [14] it was shown that the number of traffic streams associated with a priority gives more information than the QBSS load element. Although this information is not available within the standard 802.11e, wireless devices of the same vendor can still make use of this property. For this reason we also applied regression analysis with this information. The corresponding function used during regression analysis including the information about the number of traffic streams of different priorities is:

$$E[QoS] = a + bT_1 + cT_2 + dT_3 + eT_4 \quad (4)$$

where  $T_i$  is the number of stations using  $i^{th}$  priority. The results of this regression analysis is given in table 3.

**Table 2.** Linear regression analysis of QBSS load element parameters

	40%HCCA		80%HCCA		98%HCCA	
	Estimate	95% CI	Estimate	95% CI	Estimate	95% CI
a	-4.71	{-5.300,-4.120}	2.588	{2.167,3.010}	-3.205	{-4.11,-2.299}
b	-0.208	{-0.245,-0.171}	-0.195	{-0.220,-0.171}	-0.217	{-0.270,-0.164}
c	13.695	{12.835,14.555}	4.428	{3.798,5.059}	13.237	{11.876,14.598}
d	7.617	{6.717,8.518}	0.672	{-0.178,1.523}	4.146	{2.669,5.623}

**Table 3.** Linear regression analysis of number of different priorities

	40%HCCA		80%HCCA		98%HCCA	
	Estimate	95% CI	Estimate	95% CI	Estimate	95% CI
a	8.203	{7.893,8.514}	10.759	{10.563,10.955}	8.700	{8.332,9.070}
b	-2.206	{-0.270,-2.143}	-2.927	{-2.959,-2.895}	-2.417	{-2.493,-2.342}
c	-0.242	{-0.322,-0.162}	-0.209	{-0.233,-0.185}	-0.735	{-0.802,-0.669}
d	-0.201	{-0.253,-0.149}	0.000	{-0.038,0.039}	0.045	{-0.019,0.108}
e	0.058	{0.019,0.098}	0.039	{0.012,0.066}	0.003	{-0.042,0.047}

Using the estimated parameters of equations (2) and (4), we run simulations as described in section 3.2. Table 4 and 5 summarize the results.

Our first observation within tables 4 and 5 is that the percentage of true decisions decreases if the percentage of HCCA increases. This supports our claim in [14] that because of the complexity of HCCA with respect to EDCA, estimating the quality of service by only considering the QBSS load elements is very difficult. Additionally we see in both tables that having the number of traffic streams of each priority improves the percentage of true decisions in all cases. On the other hand, if we compare the results of table 4 with the results of table

**Table 4.** Percentage of true decisions made using regression analysis to find  $MOS > thresholdMOS(3.8)$

	QBSS Load Element	number of priorities
40% HCCA	77%	93%
80% HCCA	70%	80%
98% HCCA	55%	79%

**Table 5.** Percentage of true decisions made using regression analysis to find better access point out of two

	QBSS Load Element	number of priorities
40% HCCA	63%	68%
80% HCCA	60%	65%
98% HCCA	64%	64%

5, the percentage of true decisions decreases if the problem is choosing the better access point out of two instead of accepting an access point with an expected MOS value bigger than the given threshold.

Although the set of the data used during the decision making phase and the training phase of the simulations were the same for regression analysis, which brings significantly less dynamism to the problem, the best results we had using QBSS load element was 77% as seen in tables 4 and 5. On the other hand this percentage drops to 55% if nearly all the time is reserved for HCCA. Nevertheless, if the number of traffic streams of different priorities is known, the decision correctness is significantly improved and can be as high as 93%, which is an acceptable level. However the decision accuracy is very low when two access points are compared. This is mostly due to the fact that, although regression analysis can estimate the MOS for a specific state to some extent, this estimation is not exact enough to distinguish two similar states.

Considering the given results, we come to the conclusion that using simple logic such as, more channel utilization means less expected QoS for new traffic streams or correspondingly a simple way of decision making like curve fitting cannot help making reliable decisions for choosing an access point which also means that there is no straightforward relationship between the QBSS load element parameters and the resultant QoS level.

## 4.2 Results of Reinforcement Learning

The results of reinforcement learning are given in tables 6 and 7. As seen from tables 6 and 7, knowing the number of traffic streams of different priority levels allows more accurate decisions during association with an access point. Additionally although the difference is relatively lower, more time reserved for HCCA decreases the percentage of true decisions.

If we compare the performances of regression analysis and the RLA, it is clear that reinforcement learning performs in all cases better than regression analysis. In fact such kind of a result is not strange. Regression analysis tries to optimize its estimation considering all the states of the input parameters. For this

**Table 6.** Percentage of true decisions made using reinforcement learning to find  $MOS > thresholdMOS(3.2)$

	QBSS Load Element	number of priorities
40% HCCA	88%	97%
80% HCCA	81%	94%
98% HCCA	79%	95%

**Table 7.** Percentage of true decisions made using reinforcement learning to find better access point out of two

	QBSS Load Element	number of priorities
40% HCCA	84%	92%
80% HCCA	71%	94%
98% HCCA	67%	96%

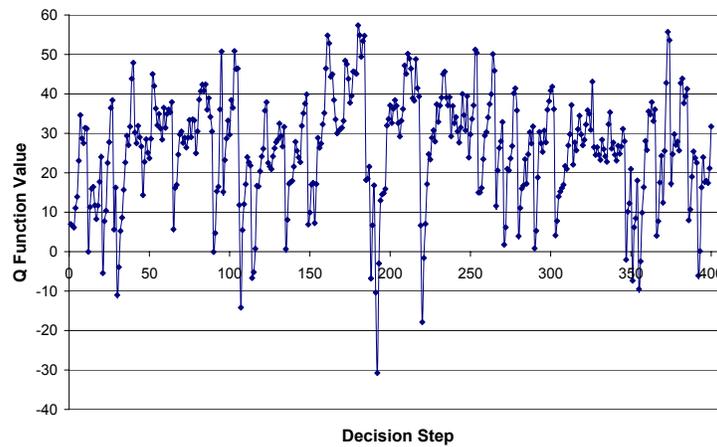
reason, especially neighboring states determine the slope of the resultant curves. Reinforcement learning deals mainly with individual states which optimizes the decision for a state apart from all other states. If the behavior of the whole system cannot be described with a convex or concave system, it is more probable that such algorithms like RLA perform better than regression analysis. During our analysis we tried many combinations of linear and non-linear equations. However, none proved to perform better than the linear regression. Although this does not mean that such an equation does not exist, alone this difficulty in finding a suitable equation representing an environment makes the use of RLA more attractive.

## 5 Deploying Reinforcement Learning over Wireless Devices

In the previous section we showed that the performance of RLA is significantly higher than regression analysis which includes but not restricted to the traditional ways of candidate access point selection algorithms. Hence it makes sense to use reinforcement learning over wireless devices of 802.11e. However reinforcement Learning is a relatively complex algorithm compared to the static way of decision making over wireless devices. For this reason it needs significantly higher hardware resources during runtime. Nevertheless we can overcome this obstacle by doing simple adjustments in the way of using reinforcement learning.

Reinforcement learning is composed of two periods, the learning period and the period in which the learned actions are used. In fact in terms of CPU time, the only challenging part is the training period since we need a high number of training steps which is in our case 1000. However it is not logical to expect that a mobile station has to wait 1000 connections in order to complete a learning procedure. Instead of that, vendors of wireless devices can give a trained Q function (See section 2.2) to the RLA. Then the problem is reduced to two subproblems. First is the decision making with an already trained Q function and the second is the adaption of the reinforcement algorithm to a new environment.

The first problem is no more different than decision making in a static way, since access points just have to check the value of a state in the Q function for making a decision. Consequently we do not have a CPU time problem. The latter is the most interesting point, since it distinguishes reinforcement learning from the static way of decision making. For this purpose we tested RLA by using a trained Q function and exposing the mobile stations to a highly dynamic environment in order to see if the Q function could be adapted with respect to new conditions successfully. Figure 3 shows the results of this analysis.



**Fig. 3.** An example to RLA adaptation within a dynamic environment. Points represent the changing Q function value of state  $s_t(2,3,74)$  at each decision step after rewarding RLA.

In figure 3, a positive Q function value means a positive answer to the access point and a negative value a negative answer. If the Q value is decreasing, then either the reward received after completing one step was not enough to compensate the decrement in the Q function value because of the discount factor  $\gamma$ , or there was a negative reward meaning that the decision was false. As it can be seen from figure 3, in nearly all of the cases where the sign of the Q function value changes from positive to negative we need less than 5 steps to change the sign. This means the correct choice is learned within less than 5 steps. In fact we consider 5 steps to be negligible for a capability of dynamic decision making. Consequently the use of reinforcement learning proves to be significantly efficient for dynamic decision making for the candidate access point selection problem.

## 6 Conclusion

The desire for a more autonomic communication environment which would work with any technology, protocol or service and without the need for intervention from the humans is growing. Such an environment should ease the way communication is being done and also maximize QoS by considering its own situation dynamically. However, the research in this field is still in its initial phase, where even a reliable and widely accepted modeling of autonomic communication is missing. On the other hand, introduction of new technologies such as 802.11e ease reaching this target, since they allow the deployment of more intelligent decision making algorithms and enable QoS negotiation among peers dynamically.

In this paper we analyzed the use of a dynamic decision making algorithm based on reinforcement learning for solving the candidate access point selection problem of 802.11e networks. We compared its results with the traditional way of decision making using regression analysis. We showed that if the stations are lacking information about network parameters which do have high correlation rates with the quality of service metrics, then only more complex algorithms like RLA can be used as a reliable solution for the candidate access point selection problem.

During our simulations, it was possible to have up to 77% correct decisions using regression analysis in case we want to know if the QoS will be over a given threshold. However this percentage fell down to 55% when we compared two access points with a relatively higher percentage of time used for HCCA. Such low levels of correct decisions showed that it is not straightforward to confirm a one to one relationship between the "QBSS load element" parameters and the quality of service metric - here MOS. On the other hand, the use of RLA which optimizes its decisions by dealing with individual states apart from the other states proved to perform much better. We reached up to 89% correct decisions if the question was finding QoS levels higher than a threshold. This result dropped to 67% when comparing QoS of two access points with higher percentage of time reserved for HCCA. As a possible enhancement to the QBSS load element, we showed that the use of the number of traffic streams of different priorities improved the performances of both algorithms considerably. Even regression analysis reached up to 90% correct decisions having only minimal differences with the performance of the reinforcement learning algorithm.

Although RLA introduces the need for more hardware resources, we showed that this is not the case if an already trained algorithm is used initially. We demonstrated that, in this way the use of the reinforcement learning algorithm during decision making can be reduced to a static way of decision making which overcomes the problem of real time application of the algorithm that might occur because of scarce hardware resources. Additionally we illustrated that RLA can adapt itself to new environments by trial and error within less than 5 false decisions within a highly dynamic environment. This introduces a substantially higher advantage when compared with the traditional way of decision making. Our next study is going to deal with the enhancement of the reinforcement

learning algorithm for admission control purposes using the traffic specification element and the ADDTS service primitive of IEEE 802.11e.

## References

1. ITU-T Rec. P.85. A Method for Subjective Performance Assessment of the Quality of Speech Voice Output Devices, 1994.
2. ITU-T Rec. G.107. The E-Model, a Computational Model for Use in Transmission Planning, 2002.
3. P. Ansel, Q. Ni, and T. Turletti. An Efficient Scheduling Scheme for IEEE 802.11e. In *Proceedings of IEEE Workshop on Modelling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, March 2004.
4. P. Ansel, Q. Ni, and T. Turletti. FHCF: An Efficient Scheduling Scheme for IEEE 802.11e. In *ACM/Kluwer Journal on Mobile Networks and Applications (MONET), Special Issue on Modelling and Optimization in Wireless and Mobile Networks*, August 2005.
5. M. Asada, E. Uchibe, S. Noda, S. Tawaratsumida, and K. Hosoda. A Vision Based Reinforcement Learning for Coordination of Soccer Playing Behaviours. In *Proceedings of AAAI-94 Workshop on AI and A-life and Entertainment*, 1994.
6. J. Baxter, A. Tridgell, and L. Weaver. KnightCap: A Chess Program that Learns by Combining TD( $\lambda$ ) with Game-tree Search. In *Proceedings of the Fifteenth International Conference on Machine Learning*, 1998.
7. Fiberlink. General Market Statistics. <http://www.fiberlink.com/release/en-US/Home/KnowledgeBase/Resources/Stats/>.
8. IEEE. \*802.11E-2005 IEEE Standard for Information technology Telecommunications and information exchange between systems Local and metropolitan area networks Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements\*, November 2005.
9. J. Koza. The genetic programming paradigm: Genetically breeding populations of computer programs to solve problems. In B. Soucek and the IRIS Group, editors, *Dynamic, Genetic, and Chaotic Programming*, pages 203–321. John Wiley, New York, 1992.
10. F. Kozamernik. Media Streaming Over the Internet an Overview of Delivery Technologies. Technical report, EBU, October 2002.
11. D. MacKay. Cambridge University Press, 2003.
12. C. Na. *IEEE 802.11 Wireless LAN Traffic Analysis: A Cross-layer Approach*. PhD thesis, The University of Texas at Austin, May 2005.
13. B. Simsek, S. Albayrak, and A. Korth. Reinforcement Learning for Procurement Agents of the Factory of the Future. *IEEE Congress on Evolutionary Computation Proceedings*, 2004.
14. B. Simsek, K. Wolter, and H. Coskun. Analysis of the QBSS Load Element Parameters of 802.11e for a priori Estimation of Service Quality. *International Journal of Simulation: Systems, Science and Technology, Special Issue: Performance Engineering of Computer and Communication Systems*, 2006.
15. R. Wiegand. *An Analysis of Cooperative Coevolutionary Algorithms*. PhD thesis, University North Carolina Charlotte, 2003.