A supervised learning approach to Cognitive Access Point Selection

Biljana Bojovic, Nicola Baldo, Jaume Nin-Guerrero and Paolo Dini
Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), IP Technologies Area
Av. Carl Friedrich Gauss 7, Castelldefels (Barcelona), Spain
{bbojovic, nbaldo, jnin, pdini}@cttc.cat

Abstract— **In this paper we present a cognitive AP selection scheme based on a supervised learning approach. In our proposal the mobile station collects measurements regarding the past link conditions and throughput performance, and leverages on this data in order to learn how to predict the performance of the available APs in order to select the best one. The prediction capabilities in our scheme are achieved by employing a Multi-layer Feed-forward Neural Network (MFNN) to learn the correlation between the observed environmental conditions and the obtained performance. Our experimental performance evaluation carried out in a testbed using the IEEE 802.11 technology shows that our solution effectively outperforms legacy AP selection strategies in a variety of scenarios.**

I. INTRODUCTION

Mobile users are often located in areas where many Access Points (APs) are available. The traditional scenario in which this fact arises is that of IEEE 802.11 WLAN, due to the popularity of this technology and the high density of deployed APs in home, enterprise and public areas. A more recent scenario with similar characteristics is that of 3G femtocells deployed with the open subscriber group paradigm, where the mobile user needs to select the femtocell to attach to among several alternatives with partially overlapping coverage areas. Depending on the propagation environment and the traffic load, the performance that the mobile user can get from each AP may vary significantly; as a consequence, it is interesting for the mobile user to identify and select the AP that will provide the best performance.

Regardless of the technology, the fundamental issue of the AP selection problem is that the performance achievable from a particular AP depends on so many environmental factors and with such complicate relationships that is not feasible to model it analytically without gross simplifications. For example, in the case of the 802.11 technology, AP schemes based solely on theoretical considerations [1]–[3] work fine only in specific situations, but fail to work properly over the big variety of conditions that are encountered in realistic scenarios, as we will show in this paper.

To solve this problem, we propose a Cognitive Networking approach [4] based on learning from the past experience. In our approach, the mobile station is equipped with a cognitive engine that learns from its past experience how the environmental conditions influence the performance of each available AP; the cognitive engine then uses this knowledge to select the AP that is expected to provide the best performance. The learning is said to be *supervised* since it is based on known training data, which in our case consists of the measurement data gathered by the mobile station in the past.

To implement these learning capabilities we use a Neural Network, which is a technique widely used for pattern recognition purposes [5]. Also the authors of [6] considered this technique in a Cognitive Radio context, and reported that it is remarkably fault tolerant. More precisely, we use use a Multi-layer Feed-forward Neural Network, which is a supervised learning technique able to model non-linear functions between inputs and outputs. We choose a MFNN because it gives a more compact model than other prediction techniques with the same generalization performance, such as support vector machines [5]. Furthermore, this choice is also supported by other previous work that considered prediction techniques applied to Cognitive Networking. For example, the authors of [7] did a study on the network traffic prediction using MFNN and Auto Regressive Integrated Moving Average, and as a result they recommend the use of a MFNN as a prediction technique with less complexity and better results.

The main contributions of this paper are 1) the description of the proposed cognitive AP selection scheme based on supervised learning, and 2) the experimental performance evaluation of its performance, carried out on the EXTREME Testbed® using the IEEE 802.11 technology. As our results will show, the proposed cognitive AP selection scheme is effective in achieving performance enhancements when compared with state of the art schemes based solely on theoretical considerations on the characteristics of the communication technology being used [1]–[3].

II. THE PROPOSED COGNITIVE AP SELECTION SCHEME

A. Cognitive AP Selection Scheme

Our primary objective is to design an AP selection scheme which will enable the mobile station to select the AP that offers the best performance. To achieve this goal, we use a cognitive scheme based on a supervised learning approach to estimate the performance for any available AP. The flow chart of the cognitive scheme is illustrated in Figure 1. First, the station performs a scan in order to discover the available access points. For each discovered AP, the station gathers the measurements regarding the environmental conditions. Then, the cognitive engine uses these measurements as inputs, and provides as output the estimated performance of that AP. This procedure is repeated for all the available APs, and depending on the estimated performance of the AP, the best AP candidate is updated. After all the APs have been evaluated, the station connects to the best AP and takes the measurements of the obtained communication performance. The environmental measurements gathered before the connection was established

978-1-4673-0040-7/11/$26.00 ©2011 IEEE

and the performance measurements gathered after connecting make up the past experience on which the cognitive engine is trained. In this way, over a period of time, the cognitive engine learns the correlation between inputs and output. Each time that the past experience is updated, the cognitive engine learns more and increases the estimation accuracy for the future predictions. As the authors of [8] discuss, this is the key aspect of the cognitive process.

Once the station is connected to the selected access point it continues to observe the environment to check whether it is convenient to switch to a different AP. This happens if the difference between the estimated throughput for the new best access point and the currently selected access point is higher than a given threshold. The threshold is needed to avoid too frequent reconnections in response to minor variations in the environmental conditions.

B. Considerations on learning

In real devices, learning from the past experience is expected to be implemented either by using experience gathered in the laboratory (e.g, training the device before it is sold) or by gathering experience during the ordinary usage in real life scenarios. The main benefit of the first approach is that the AP selection scheme relies on prior knowledge since its first usage, but this knowledge gathered in laboratory does not include non-ideal behavior which can happen in reality. On the other hand, in the second approach knowledge is gathered from the real life scenarios, but since the mobile station starts using the AP selection scheme without having prior knowledge, it will likely make wrong AP selection decisions until it gathers enough experience. We suggest that the best approach is to install on the mobile station the AP selection scheme with a cognitive engine that is pre-trained in the laboratory, and then to increase knowledge during real life usage.

C. Specialization of the Cognitive Scheme for 802.11 networks

For the implementation of the cognitive engine we use a two-layer[1] MFNN topology, since that topology has the minimum number of hidden layers which is sufficient to approximate any continuous function on a compact input domain to arbitrary accuracy, provided the network has a sufficiently large number of hidden units [5]. For the inputs and the outputs of the MFNN we choose metrics which are representative of the particular scenario being considered. In particular, for the MFNN inputs we use several metrics that represent the distance from considered AP and its background traffic load. The metrics that are used as MFNN inputs are the following:

- the signal to noise ratio $\gamma \in \mathcal{R}$
- the probability of failure $p_f \in [0,1] \subset \mathcal{R}$
- the business ratio $b_r \in [0,1] \subset \mathcal{R}$
- the average beacon delay $t_b \in \mathbb{R}_{\geq 0}$
- the number of detected stations $n_s \in \mathcal{Z}$

γ is defined as the ratio of the received signal power (RSSI) to the noise power corrupting the signal; both measurements are available on most commercial WLAN devices.

[1]We use the same terminology as in [5] where the number of layers refers to the number of layers of adaptive weights.

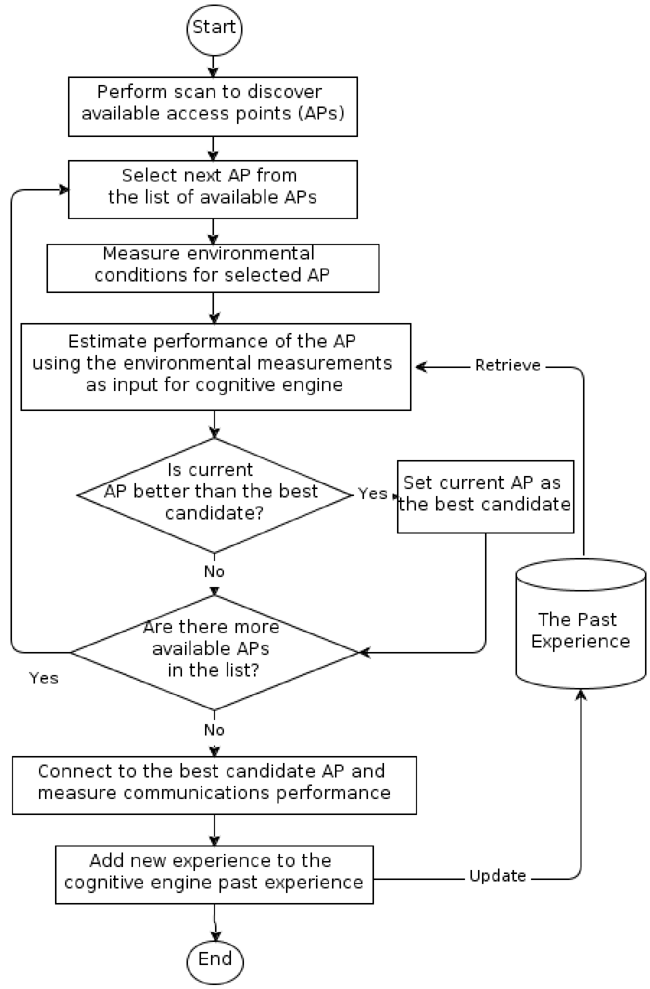

Fig. 1. The cognitive scheme flow chart

p_f represents the ratio of the number of retransmitted frame exchanges to the total number of frame exchanges. We calculate p_f using the $retry$ frame flag from the MAC header of IEEE 802.11. The $retry$ flag has value of 0 when a frame is transmitted for the first time and value of 1 when the frame is retransmitted. Let s and r be the numbers of DATA frames which are successfully decoded by the mobile station and which have the value of the retry flag 0 and 1, respectively. We define p_f as:

$$p_f = r/(r+s) \tag{1}$$

b_r represents the ratio of time in which the channel is occupied by the frame transmissions that the mobile node can decode successfully. The calculation of b_r is done in the following way:

$$b_r = \frac{\sum_{i=1}^{n} T_i}{T} \tag{2}$$

where T is duration of the time interval in which the mobile node gathers measurements, n is the total number of frame exchange sequences which are transmitted during the interval T and T_i is duration of the $i-th$ frame exchange sequence. T_i is calculated as the sum of the duration of the DATA frame, the ACK frame and the DIFS (or AIFS in the case of QoS

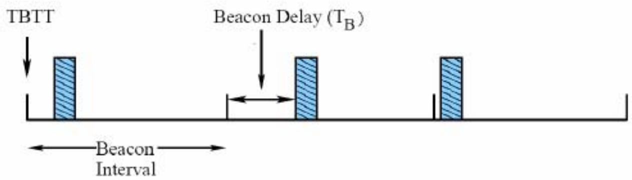

Fig. 2. Beacon transmission delays and TBTT on an AP

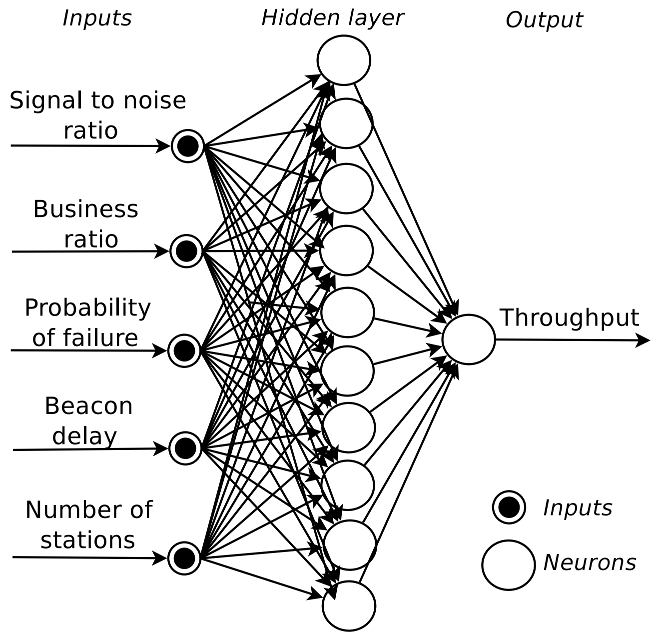

Fig. 3. Architecture of the MFNN with five inputs, a hidden layer and one output

support) and SIFS, as is defined by the standard [9].

t_b is the average beacon delay in the beacon transmissions from an access point during the time T in which the mobile node gathers the measurements. The beacon delay is equal to the difference between the timestamp when beacon is transmitted and target beacon transmission time (TBTT), as is illustrated in Figure 2. The value of the beacon timestamp is obtained from the management frame field and TBTT is calculated and updated using the beacon interval time and beacon sequence numbers, which are obtained from the beacon frame.

n_s is the number of stations which are detected by the considered station to be exchanging frames with the AP. This metric is calculated by counting the distinct values of the source and the destination address fields in the decoded frames. Since we consider scenarios in which one dedicated orthogonal channel is used for each AP, we can calculate n_s by counting all the different station addresses which are seen by the monitoring station.

For the MFNN output (the performance metric) we choose the downlink throughput of TCP, since it is representative of the operations most commonly performed by the mobile users in a WLAN (e.g., web browsing, file transfer). Using the previously defined MFNN inputs and output we get the architecture for MFNN architecture which is shown in Figure 3.

III. Performance Evaluation

A. Setup of the EXTREME Testbed

We carried out an experimental performance evaluation of the proposed scheme using the EXTREME Testbed® [10]. The configuration that we used for experiments in the testbed is shown in Figure 4. All stations are regular PCs running the Linux operating system, using the madwifi driver for the wireless interface, and they are connected via Ethernet to another PC which is used as a central experiment controller. We use the central controller PC to run the experiments, to monitor their execution and to gather the measurements from all the stations. The wireless interfaces of all the stations are interconnected using RF cables to minimize noise-related channel errors. We used a commercial CISCO Aironet 1200 Series access point configured in 802.11g mode. The settings of the access point are shown in Table I. On each station we installed the Iperf tool (version 2.0.4) [11] to create TCP data streams in the downlink and to measure the obtained throughput; more information on the Iperf tool can be found in [12]. One station, referred to as the *tagged* station, is the node running the AP selection scheme, which is connected to the AP via attenuator. The other stations are referred to as *background* stations; they are not attenuated, and hence they ideally can use the highest transmission rate of 802.11g. For all stations, the actual rate being used is dynamically selected by madwifi's default rate adaptation algorithm.

The experiments are run using a number of active background stations $N \in \{1, 2, 3, 4, 5, 6, 7, 8\}$, and using attenuator value $t \in \{0, 5, 10, 15, 20, 25\}$ dB; we note that 0 dB corresponds to the station being able to use the maximum physical rate, and 25 dB corresponds to the station starting to loose connection with the AP. In each experiment we fix the number of active background stations N and the attenuator value t. During each experiment the active background stations perform a TCP downlink transfer using the Iperf tool. In the first part of the experiment, the tagged station monitors the wireless channel and gathers the measurements in PCAP traces using its wireless interface in monitor mode. In the second part of the experiment, the tagged station connects to the AP, performs a TCP file download and measures the obtained throughput using the Iperf tool.

We ran 48 different experiment configuration changing N and t; for each configuration we ran experiments using 8 different permutations of the available machines from the EXTREME Testbed®, in order to minimize the eventual bias of the measurements on the specific hardware being used. For each setup we run 4 independent repetitions which gives a total of 1536 samples.[2]

After the experiments are done, we use the PCAP Trace Parser tool [13] to process the PCAP traces and to calculate the metrics described in Section II.B. Since the PCAP parser is not providing the beacon delay as output, we developed another software which uses the PCAP trace to calculate the average beacon delay for the whole measurement interval. The Iperf tool provides as output value the measured throughput, so there was no need for additional processing of the throughput measurements. After all the experimental results are processed,

[2]One sample represents a set of inputs and its related outputs.

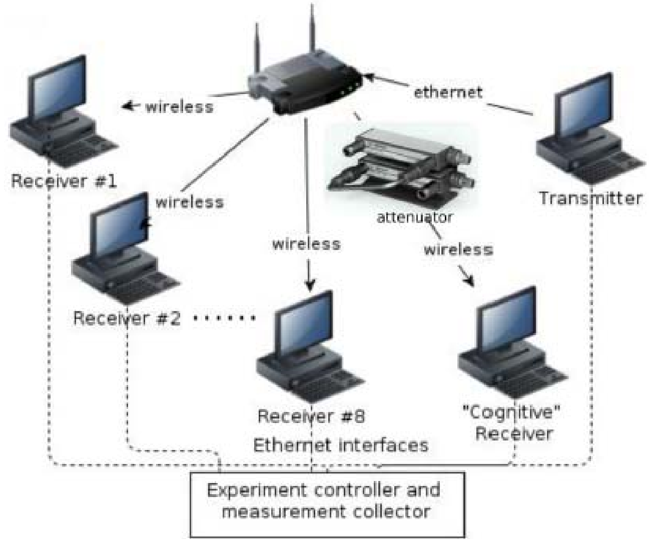

Fig. 4. Setup of the EXTREME Testbed

TABLE I
ACCESS POINT SETUP

Network interface	802.11g
Operational Rates	1.0, 2.0, 5.5, 6.0, 9.0, 11.0, 12.0, 18.0, 24.0, 36.0, 48.0, 54.0 Mb/sec
Radio Channel	2412 MHz Channel 1
Short Slot-Time	Enabled
Beacon period	51200 μ

all the obtained data is stored in a MYSQL database, from where it will be fetched for the training and the testing the Neural Network.

The performance evaluation is done in two phases. In the first phase we identify the values of the training parameters that provide the best MFNN performance; in the second phase we evaluate the actual AP selection scheme using the experimental results just described. These phases are explained in the next subsections.

B. The Neural Network Training

The training of the Neural Network is done based on the experimental results described in Section III.A. The samples are divided in two sets of equal size, where the first is used for the training of the MFNN and the second is used for the testing. For the training, the environmental and throughput measurements are used, respectively, as inputs and output for the Neural Network. In this way, the MFNN learns how the throughput depends on the environmental conditions measured before the connection with the AP is established.

For the implementation of the MFNN we used FANN, which is a publicly available software library [14]. Following the recommendations from [15], we set up FANN to use the iRPROP- batch training algorithm [16]. We tried different values of the MFNN configuration parameters, and then we chose the parameters for which our MFNN gave the best performance. The parameters which we varied during the training of the MFNN are:

- the number of nodes in the hidden layer $H \in \mathcal{Z}$
- the maximum number of epochs $E \in \mathcal{Z}$
- the learning rate $L \in [0, 1]$

For the description of the meaning of these parameters the reader is referred to [17]. The testing of the MFNN consists in calculating, for each sample in the testing set, the error between the throughput value predicted by the MFNN (evaluated with the environmental measurements as inputs) and the throughput measured by Iperf which is also recorded in the sample. In Figure 5 we report the Normalized Root Mean Squared Error (NRMSE) of the estimated throughput, which is calculated in the following way:

$$NRMSE = \frac{RMSE}{X_{max} - X_{min}}, \tag{3}$$

where RMSE is the Root Mean Squared Error calculated on the whole testing set, and X_{max} and X_{min} are, respectively, the maximum and the minimum values of the measured throughput. As evident from the figure, the most important parameter for the training of the MFNN is E, while H and L do not appear to play an important role. The best performance of the MFNN is achieved with $E = 70$, for which the NRMSE has its minimum value equal to 0.0835. For values of E lower than 70, the error is bigger because the MFNN does not have enough knowledge yet. Conversely for values higher than 70, the MFNN becomes over-trained and too specialized on the training data set; in other words it looses generalization, which is needed for the good interpretation of the testing data set. Since the MFNN provides the best estimation for $E = 70$, we will use this value for the training of the MFNN for the AP selection scheme.

The interval during which the station gathers measurements is configurable. Regardless of the AP selection scheme being used, the measurement time influence the measurement accuracy. To get a high accuracy, especially for the load based schemes [2], [3], it is set to a few seconds. This duration is at least 3 orders of magnitude higher than the time necessary for the MFNN to provide its prediction. Similarly, the time needed to accumulate the past experience is orders of magnitude greater than the time needed to run the iRPROP batch training algorithm. For this reason, we consider that the computational time of using an MFNN for AP selection is negligible.

C. Evaluation of the Cognitive AP selection scheme

For the performance evaluation of the cognitive AP selection scheme we consider a scenario in which the mobile station is in coverage of two[3] APs, respectively AP1 and AP2. In general the APs have different numbers of background stations, respectively N_1 and N_2. The values of the signal to noise ratio that the tagged station receives from the APs are in general different and respectively denoted as γ_1 and γ_2. Since running experiments with two APs is cumbersome due to the many combinations of N_1, N_2, γ_1 and γ_2, we recreate this scenario artificially by using the Cartesian product of two subsets of the results obtained from the experiments with a single AP. Let S be the result set of the single AP experiments, and let $s = (N, \gamma, r)$ be the generic element of S, where r is

[3]Even if we do performance evaluation on scenario with two AP, our scheme can be applied to any number of APs.

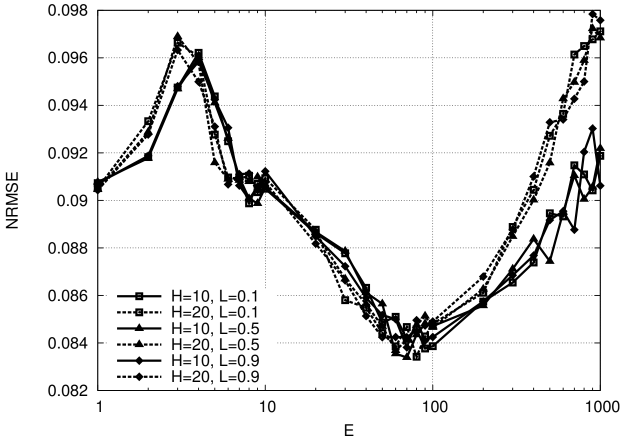

Fig. 5. Performance of the MFNN based estimation of the AP throughput

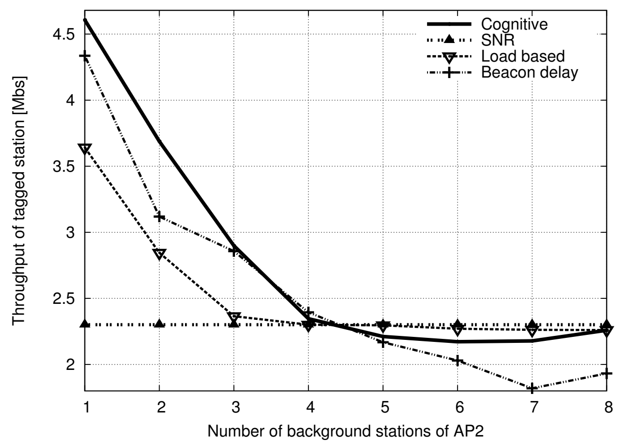

Fig. 6. Performance of different AP selection schemes in scenario with two AP only, for different values of N_2

TABLE II
EXPERIMENTS CONFIGURATION SETTINGS

	Scenario name		
	Varying N	Varying γ	General
$N_{1min} - N_{1max}$	5-5	1-1	1-8
$N_{2min} - N_{2max}$	1-8 (N)	4-8	1-8
$\gamma_{1min} - \gamma_{1max}[dB]$	50-60	50-55	35-60
$\gamma_{2min} - \gamma_{2max}[dB]$	35-50	35-60 (γ)	35-60

the repetition identifier. Then, we define two subsets of S in the following way:

$$S_1 = \{s : N_{1min} \leq N_1 \leq N_{1max}, \\ \gamma_{1min} \leq \gamma_1 \leq \gamma_{1max}, r < r_{thr}\} \tag{4}$$

$$S_2 = \{s : N_{2min} \leq N_2 \leq N_{2max}, \\ \gamma_{2min} \leq \gamma_2 \leq \gamma_{2max}, r \geq r_{thr}\} \tag{5}$$

The sample from the Cartesian product $S_{(1\times2)} = S_1 \times S_2$ is the ordered pair (s_1, s_2), where s_1 and s_2 are respectively elements from S_1 and S_2, and represent the performance of AP1 and AP2 respectively. In Table II the subsets constraint values are shown for the different scenarios with 2 APs that we consider.

1) Scenario 1(Varying N_2): In order to analyze the performance of the cognitive scheme for the different load of the APs, we consider the experiments with a fixed N_1, and with γ_1 and γ_2 varying in narrow intervals, while we vary N_2 for all the values of N. The settings for all the experiments which are run for this scenario are in Table II. We compare our solution with the scheme based on the signal to noise (SNR), the load based scheme and the beacon delay scheme. The load based scheme uses the business ratio as selection criteria, while the beacon delay scheme uses average beacon delay. In Figure 6 we show a comparison of the performance of these AP selection schemes for the scenario described above. The X axis represents N_2, and the Y axis represents average throughput which is obtained by the tagged station using each AP selection scheme. Since AP2 offers a stronger γ, AP2 is expected to be a better choice when $N_2 < N_1$, because it is less loaded and closer. In the case when $N_2 > N_1$, AP1 is expected to be better choice because it is less loaded. Thus, the expected behavior for all schemes is that the throughput obtained by the tagged station decreases with increase of the load on AP2. In this scenario, since the SNR scheme neglects the load of APs, fraction of correct decisions remain constant for any value of N_2. The load based AP selection, which performs selection using the business ratio, in this scenario has good performance, but still in many cases it makes wrong decisions because it does not consider γ. The AP selection scheme based on the beacon delay has similar performance to the load based scheme, but for $N_2 > N_1$ in many cases it selects wrong AP, because the average beacon delays for the AP1 and the AP2 start to have very similar values. Our scheme gives very good performance for all values of N_2. Due to possible errors in the estimation described in Section III.B the Cognitive scheme sometimes selects a wrong AP, especially when both AP have a similar load and a similar γ is detected by the station. However, the cognitive scheme shows a robust behavior and it chooses good AP much more often than the other schemes as is shown in Figure 8, in which this scenario is denoted as “Scenario1”.

2) Scenario 2 (Varying γ_2): The second scenario is characterized by a fixed N_1, N_2 and γ_1 varying in a narrow interval, while γ_2 is varying in the whole interval of γ. We choose this scenario in order to evaluate the performance of the cognitive AP selection scheme when the detected values of γ from AP1 and AP2 are very different. Also the settings for this scenario are summarized in Table II. In Figure 7 we show the performance of our scheme, compared with the SNR scheme, the load based scheme and the beacon delay scheme. As we expected the SNR scheme has good performance when $\gamma_1 > \gamma_2$, since it selects AP1 which in all experiments is less loaded. Conversely, for $\gamma_1 < \gamma_2$ the SNR scheme performs worse because it will always choose AP2 which is more loaded. The load based scheme and the beacon delay scheme provide more stable performance than the SNR; also, those schemes cannot detect high differences between γ_1 and γ_2 which are present in this scenario, and react appropriately to this condition. We notice that the cognitive scheme has very good performance and stable behavior in AP selection for all

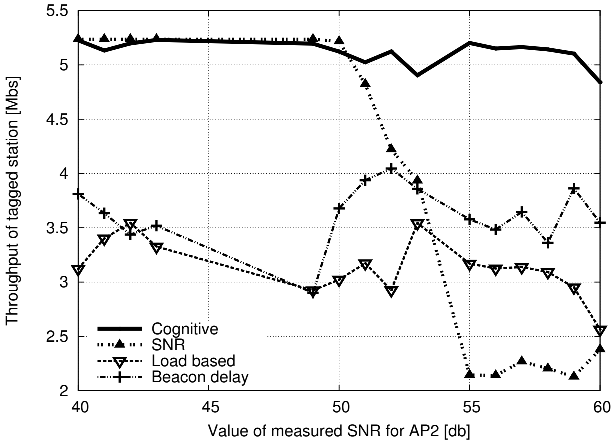

Fig. 7. Performance of different AP selection schemes in scenario with two AP, for different values γ_2

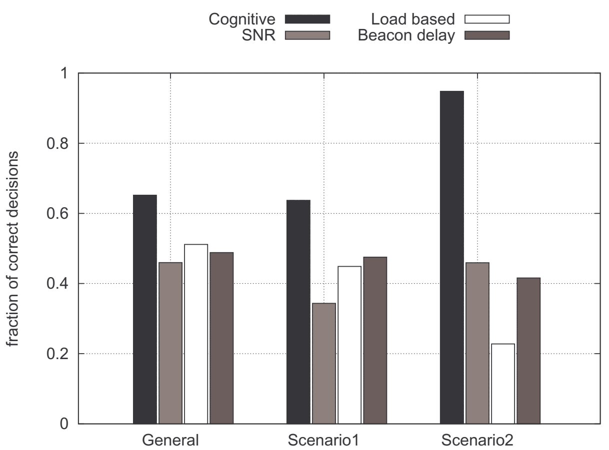

Fig. 8. Histogram for different algorithms

values of γ_2, as we can see from Figure 7 and 8, in which this scenario is denoted as "Scenario2".

3) General scenario: We already discussed that, in cases with significant difference in N and γ, the cognitive scheme has very good performance. In order to evaluate the performance in a more generic way we choose the subsets for the Cartesian product so that we involve all possible scenarios. To achieve this we do not enforce any constraint on N_1, N_2, γ_1 and γ_2, as we show in Table II. We use r to divide results of the single-AP experiments in two exclusive subsets, from which we randomly select a limited number of experiments (1000) which will be used in the Cartesian product. As a result we get 100000 two-AP experiments for this scenario. In Figure 8 we can see that in this general scenario the cognitive scheme performs better than the other schemes, with an average of +15% correct decisions.

IV. Conclusions

In this paper we proposed a cognitive AP selection scheme based on learning from past experience. Our scheme uses an MFNN for the implementation of its learning capabilities and aims at selecting the AP that is expected to yield the best throughput according to the past experienced performance. We described our experimental performance evaluation study done in a testbed using the IEEE 802.11 technology, where we compared our cognitive solution with state of the art schemes known from the literature. Our results show that the proposed cognitive AP selection scheme achieves a remarkable performance improvement with respect to the state of the art schemes, especially in presence of highly variable environmental data. We conclude that a cognitive approach to AP selection based on learning techniques is more effective than traditional solutions in addressing the complexity of today's communication systems and the variability and unpredictability of real scenarios.

V. Acknowledgements

This work was partially supported by the Spanish Ministry of Science and Innovation under grant number TEC2008-06826/TEC (project ARTICO), by the Catalan Regional Government under grant 2009SGR-940, by COST Action IC0902 "Cognitive Radio and Networking for Cooperative " and by the FP7-ICT Network of Excellence ACROPOLIS "Advanced coexistence technologies for radio optimization in licensed and unlicensed spectrum".

References

[1] Y. Fukuda, T. Abe, and Y. Oie, "Decentralized access point selection architecture for wireless LANs," in *Wireless Telecommunications Symposium (WTS), 2004*, May 2004.

[2] Jae-wook Jang and Yeon-sup Lim and Chong-kwon Kim, "Traffic-aware decentralized AP selection for multi-rate in WLANs," in *Proceedings of International Conference on Advanced Communication Technology (ICACT)*, February 2010.

[3] J.-C. Chen, T.-C. Chen, T. Zhang, and E. van den Berg, "Effective AP Selection and Load Balancing in IEEE 802.11 Wireless LANs," in *Proceedings of Global Telecommunications Conference (GLOBECOM)*, December 2006.

[4] C. Fortuna and M. Mohorcic, "Trends in the development of communication networks: Cognitive networks." *Computer Networks*, 2009.

[5] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[6] C. Doerr, D. Sicker, and D. Grunwald, "Experiences Implementing Cognitive Radio Control Algorithms," in *Proceedings of Global Telecommunications Conference (GLOBECOM)*, November 2007.

[7] H. Feng and Y. Shu, "Study on network traffic prediction techniques," in *Proceedings of Wireless Communications, Networking and Mobile Computing (WiCOM)*, September 2005.

[8] R. W. Thomas, D. H. Friend, L. A. Dasilva, and A. B. Mackenzie, "Cognitive networks: adaptation and learning to achieve end-to-end performance objectives," *Communications Magazine, IEEE*, December 2006.

[9] IEEE Std. 802.11-2007, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," June 2007.

[10] M. Portoles-Comeras, M. Requena-Esteso, J. Mangues-Bafalluy, and M. Cardenete-Suriol, "EXTREME: combining the ease of management of multi-user experimental facilities and the flexibility of proof of concept testbeds," in *Proceedings of Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom)*, 2006.

[11] "The IPERF network testing tool for TCP and UDP data streams." [Online]. Available: http://iperf.sourceforge.net/

[12] E. Yildirim, I. Suslu, and T. Kosar, "Which network measurement tool is right for you? a multidimensional comparison study," in *Proceedings of International Conference on Grid Computing (GRID)*, October 2008.

[13] "Pcap Trace Parser." [Online]. Available: http://iptechwiki.cttc.es/Parser

[14] "Fast Artificial Neural Network Library (FANN)." [Online]. Available: http://leenissen.dk/fann/

[15] S. Nissen, "Neural Networks made simple," 2005. [Online]. Available: http://fann.sf.net/fann_en.pdf

[16] C. Igel and M. Hüsken, "Improving the RPROP learning algorithm," in *Proceedings of the ICSC Symposium on Neural Computation*, May 2000.

[17] B. Wilson, "The Machine learning dictionary." [Online]. Available: http://www.cse.unsw.edu.au/~billw/mldict.html