A Neural Network Based Cognitive Engine for IEEE 802.11 WLAN Access Point Selection

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

Abstract— **Nowadays IEEE 802.11 WLANs are widely deployed; in spite of this, the issue of designing an efficient and practical Access Point selection schemes that can provide the best throughput performance in a variety of link conditions is still open. In this paper we present a Cognitive AP selection scheme that allows the mobile station to learn from its past experience how to select the best AP. In our proposal the mobile station collects measurements regarding the past link conditions and throughput performance, and a cognitive engine based on a Neural Network trained on this data drives the AP selection process. Our performance evaluation shows that the proposed scheme has very good performance in a variety of scenarios, as opposed to other algorithms previously proposed in the literature which perform well only in specific cases and cannot address the non-idealities typical under real conditions.**

I. Introduction

In this paper we deal with the problem of Access Point (AP) selection in IEEE 802.11 Wireless LANs (WLANs). Mobile users are often located in areas where many APs are available; typical scenarios are residential and enterprise buildings, university campuses and airports. Depending on the propagation environment and the traffic load, the throughput that the mobile user can get from different APs may vary significantly; as a consequence, it is interesting for the mobile user to identify and select the AP that will provide the highest throughput.

Since the IEEE 802.11 standard [1] does not specify how to select the most appropriate AP, manufacturers implement different solutions. In fact, in order to be competitive in the market, it is interesting for a manufacturer to implement an AP selection scheme which will provide good throughput performance to the mobile user. The scheme most commonly implemented in mobile devices is based on the Received Signal Strength Indicator (RSSI) measurement and consists in selecting the AP from which the device receives beacon frames with the strongest signal strength. This type of scheme is easy to implement and does not require neither changes on the APs nor additional exchange of control information. However, its major drawback is that the APs which are close to the majority of the users can become overloaded, while other APs which are at a longer distance remain underutilized. This is due to the fact that the RSSI scheme is neglecting the different traffic load of the available APs.

To address this problem, several new schemes have been proposed in the literature that consider traffic load for the AP selection process.These schemes can be divided into two main groups: centralized and decentralized. In centralized schemes, a separate management system connected to the WLAN provides the mobile users with information regarding the load of the APs; such a scheme is proposed in [2]. The benefit of this scheme is that it has all the information which is necessary to perform load balancing among the APs. A major issue of this approach is that it can be used only at locations where this management system is deployed, and therefore it does not allow an efficient AP selection in legacy WLAN deployments. For this reason we do not consider centralized schemes in this paper.

Decentralized load based AP selection schemes, on the other hand, are entirely implemented on the mobile stations and do not require changes to neither AP nor the WLAN backbone. These schemes use different metrics, which are representative of the load on each AP. In [3] the authors use the probe delay metric which is defined as the difference between the probe request time and the probe response time for a specific AP. The problem with this algorithm is that it only considers those APs whose Signal to Noise Ratio (SNR) is greater than a threshold, and excludes others which could give better performance in situations where the APs with the higher SNRs are overloaded. Moreover, since the probe delay mostly depends on the uplink traffic, the schemes based on this metric will give bad results when most of the traffic is downlink (e.g. web browsing). In [4] and [5] the authors propose a schemes based on the average transmission time metric for the estimation of the throughput. Since the average transmission time is calculated using measurements gathered by monitoring the wireless medium, and since the mobile station's coverage region is different from that of the AP, the mobile station cannot always decode all the frames exchanged between the AP and other stations. Hence, in the presence of hidden nodes, the estimated value of the throughput becomes less accurate, potentially yielding incorrect AP selection decisions. For this reason, some authors, e.g. in [4] and [6], suggest modifications in an AP in order to provide the stations with additional information about the AP load.

In order to design an AP selection scheme which will provide better performance with respect to the state of the art, we propose a decentralized scheme which uses both the RSSI and load based metrics. Because of the complexity of designing an analytical model based on these metrics that can perform well also in realistic conditions (e.g in presence of hidden nodes), we choose a cognitive approach: the mobile station learns from its past experience how the environmental conditions influence the throughput performance. Our scheme uses a Neural Network to implement these learning capabilities. After learning has been accomplished, the mobile station uses the acquired knowledge to estimate the throughput of all available APs, and then selects the AP with the highest value of the estimated throughput. As we will show in our performance evaluation section, the proposed AP selection scheme achieves a significant performance enhancement with respect to the RSSI and load based decentralized AP selection schemes.

II. Proposed scheme

A. System description

Our primary objective is to design an AP selection scheme which will enable the mobile station to select the AP with the highest value of the throughput. To achieve this goal, our approach is to implement a cognitive engine which estimates the throughput for any available AP. Our scheme uses only measurements which are gathered by the mobile station monitoring the wireless medium, and therefore it can be installed on any mobile station, without requiring any special capability from the wireless network infrastructure.

The key aspect of our scheme is that an accurate estimation of the throughput is achieved by learning from past environmental conditions and throughput measurements. In real devices, this is expected to be done either by using experience gathered in the laboratory (e.g, training the device before it is sold) or gathering experience during normal usage. The first approach is better because the AP selection schemes can rely on some prior knowledge since its first usage, but on the other hand knowledge gathered in laboratory does not include non-ideal behavior which can happen in reality. The other approach is to start the usage of the AP selection scheme on the mobile station without any prior knowledge, and then do all the training using experience gathered during the normal usage. The drawback of this approach is that the mobile station starts using the AP selection scheme without knowledge and it will likely make wrong decisions for AP selection until it gathers enough experience. On the other hand, using this approach the device is trained with the experience from real life scenarios, where all irregularities are included. 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 gather new experience from the real life usage in order to increase its knowledge and to improve the estimation of the throughput.

Once the training is completed the mobile station uses the trained cognitive engine to determine the expected throughput for all available APs. The mobile station then compares these values and selects the AP with the highest value of the estimated throughput.

B. Technical specification

For the implementation of the cognitive scheme we use Feed Forward Neural Network (FFNN), which is a learning technique able to model non-linear functions between inputs and outputs. We choose a FFNN because it gives a more compact model than other prediction techniques with the same generalization performance, such as support vector machines [7]. We use a two-layer[1] FFNN which contains minimal sufficient number of hidden layers necessary to approximate any continuous function on a input domain to an arbitrary accuracy, provided the network has a sufficiently large number of hidden units [7]. For the inputs and the outputs of the Neural Network, we use metrics that represent, respectively, environmental conditions and the throughput that the mobile station can achieve by connecting to that AP. Since the coverage and interference regions of the mobile station and the AP are, in general, different, it is obvious that the station cannot gather perfect environmental information regarding the AP. Therefore, the environmental measurements will represent how the mobile station "sees" the communication environment.

For the environmental measurements we use the following metrics:

- 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 number of detected stations $n_s \in \mathcal{Z}$

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

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:

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

we note that, when rate adaptation is being used, p_f is mostly representative of MAC collisions.

The third metric, 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 support) and SIFS, as is defined by the standard [1].

The fourth metric, n_s, represents 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.

Using the previously mentioned metrics as the inputs for the Neural Network, and the application layer throughput as the output, we get the architecture for the Neural Network Based Cognitive Engine which is shown in Figure 1.

[1]We use the same terminology as in [7] where number of layers refers to the number of layers of adaptive weights.

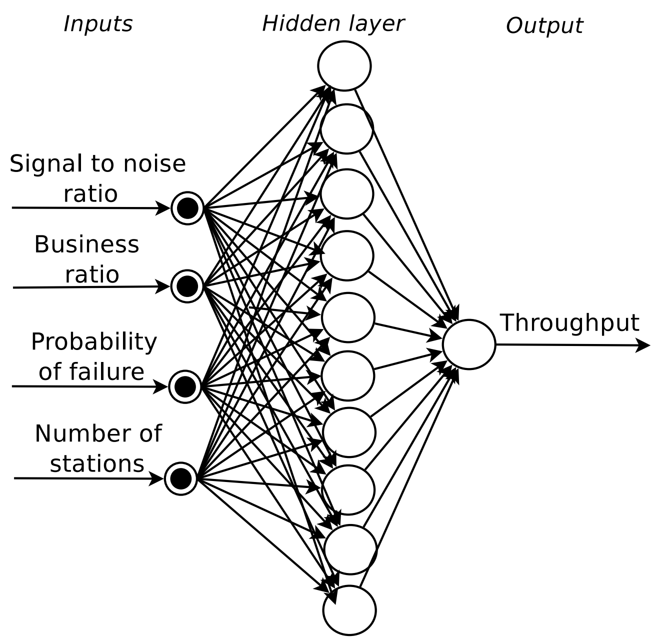

Fig. 1. Architecture of Neural Network Based Cognitive Engine with four inputs, a hidden layer and one output

III. Performance Evaluation

A. Evaluation setup

For the performance evaluation of the proposed scheme we used the ns-3 network simulator [8]. We chose ns-3 because it is open source, it has good TCP/IP and WIFI models (see for example the validation study in [9]), and it has very good run-time performance and memory usage [10]. Moreover, ns-3 supports saving packet traces in the PCAP format with RADIOTAP header, which is the same format normally used for real wifi devices. We consider this feature important to make it easy to implement the proposed scheme in a real testbed in the future. The version of the simulator that we used is 3.9. For the processing of the PCAP traces and the calculation of the metrics described in Section II.B we use the PCAP Trace Parser [11], which we developed at the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC). The data obtained from the parser is stored in a MYSQL database, from where it is fetched for training and testing the Neural Network. For the implementation of the Neural Network we used FANN, which is a publicly available software library [12]. Following the recommendations from [13], we use the iRPROP- batch training algorithm [14]. The performance evaluation of our scheme is done in two parts. In the first part we train the Neural Network and identify the values of the training parameters that provide the best throughput estimation performance. Then, in the second part, we evaluate the AP selection scheme.

B. Neural Network training

In general, the accuracy of a Neural Network depends on the parameters which are used for its training [15]. To discover the values of the training parameters for which our Neural Network gives the best performance we carried out different training sessions varying the following parameters:

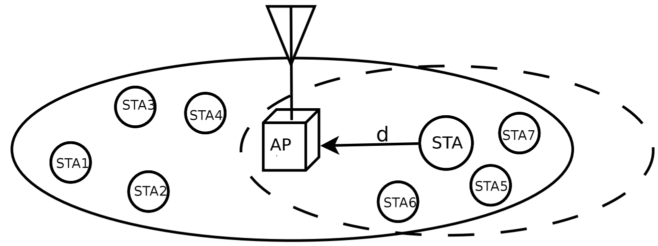

Fig. 2. Neural Network training scenario

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

The training of the Neural Network is done with measurements obtained in a scenario in which one mobile station, which in the remainder of the paper is referred to as the tagged station, is in coverage of a single AP. A number N of other stations, called background stations, are connected to the AP. An example of this scenario is represented in Figure 2 where we have one AP with 7 background stations (STA1-STA7) plus the tagged station (STA), which is at a distance d from the AP. We fix the number and position of the stations for each experiment. We run a TCP file download on all the stations. The background stations are uniformly randomly distributed within a disc centered at the AP and having radius equal to 150 meters. We chose a radius equal to 150 meters because in ns-3 this is approximately the radius of coverage area of an AP using the default parameters. For rate adaptation we chose the algorithm which in ns-3 is called "IdealWifiManager". This algorithm is similar to Receiver-Based AutoRate (RBAR) [16], where each transmitter keeps track of the last SNR sent back by a receiver and uses it to pick a transmission mode based on a set of SNR thresholds built from a target Bit Error Rate (BER) and transmission mode-specific SNR/BER curves. During each experiment, the behavior of the tagged station is a bit different from that of background stations. In the first part of the experiment the tagged station is monitoring the wireless medium and gathers the measurements which are necessary for the calculation of the metrics described in Section II.B, but does not perform any data communication. In the second part the tagged station connects to the AP, performs a TCP file download and measures the throughput of this flow. The gathered environmental and throughput measurements are used, respectively, as inputs and outputs for Neural Network training. In this way, the Neural Network will learn how the throughput depends on the environmental conditions measured before the connection with the AP is established.

We ran 192 different experiments changing N and d, and for each configuration we ran 30 independent repetitions, which gives in total 5760 samples.[2]. The 5760 samples are divided in two sets of equal size, where the first is used for the training of the Neural Network and the second is used for the testing. The testing consists of calculating the error between the throughput value predicted by Neural Network and the actual value measured by the device. In Figure 3 we report the Normalized Root Mean Squared Error (NRMSE) of the

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

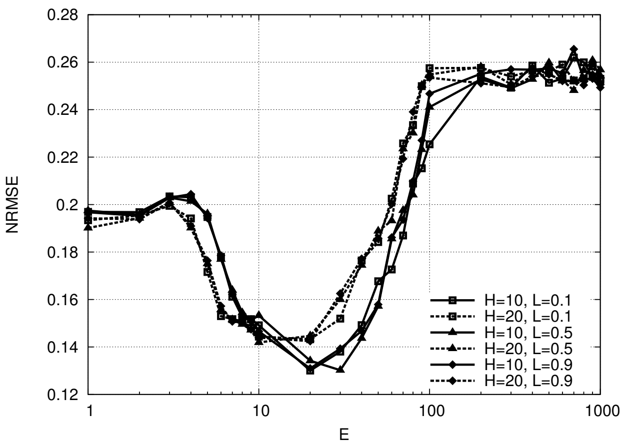

Fig. 3. Performance of Neural Network based estimation of AP throughput

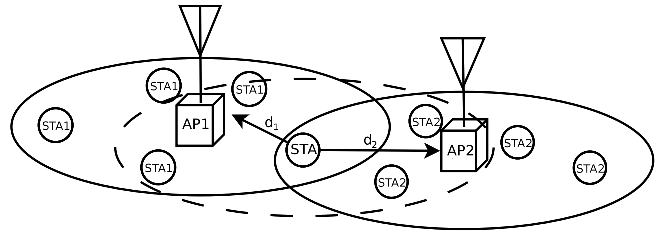

Fig. 4. Tested scenario for AP selection

TABLE I
EXPERIMENTS SETTINGS

Scenario	N_1	N_2	$d_1[m]$	$d_2[m]$
Varying load	5	1-11	120	30
Varying distance	2	11	80	0-150

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 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 Neural Network is E, while H and L do not appear to play an important role. The best performance of the Neural Network is achieved with E=20, for which the NRMSE has its minimum value. For values of E lower than 20, the error is bigger because the Neural Network does not have enough knowledge yet; conversely for values higher than 20 the Neural Network becomes over-trained and too specialized on the training data set, so it looses generalization, which is needed for the good interpretation of the testing data set. Since the Neural Network provides the best estimation for E=20, we will use this value for the training of the Neural Network used in performance evaluation of the AP selection scheme, which we will describe in the next subsection.

C. Evaluation of the AP selection scheme

For the performance evaluation of our AP selection scheme we consider a scenario in which the tagged station is in coverage of two[3] APs, respectively called AP1 and AP2, operating on orthogonal channels. This is illustrated in Figure 4. In this scenario the APs have in general a different numbers of background stations, respectively N_1 and N_2, and are at a different distance from the tagged station, respectively d_1 and d_2. For each experiment we fix the number of background stations and the positions of all stations and APs. The background stations for each AP are uniformly randomly distributed within a disc of radius 150 meters centered on the AP, similarly to the scenario described in section III.B. On each background station a TCP file download is performed. The tagged station is passively monitoring the wireless channel for the purpose of calculating the metrics that are described in Section II.B. We used the Neural Network trained with the training parameters and data described previously in section III.B. The Neural Network then provides the value of the estimated throughput for both AP1 and AP2. After getting these values the tagged station selects the AP with the higher value of estimated throughput, connects to it and starts a TCP file download. The experiment is repeated with different values of N_1, N_2, d_1 and d_2.

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

1) Scenario with varying load: In order to present how the performance of our scheme depends on the difference in the load of the APs, we first consider the subset of experiments with a fixed value of d_1, d_2 and N_1, while we vary N_2. The values of the parameters are summarized in table I. We compare our cognitive AP selection strategy with the RSSI and the load based schemes. In Figure 5 we show a comparison of the performance of all the AP selection schemes for the scenario described above. The X axis represents N_2, and the Y axis represents the average throughput which is obtained by the tagged station using each AP selection scheme. Since AP2 is closer to the tagged station, AP2 is a better choice when $N_2 < N_1$, because it is less loaded and closer. Conversely, when $N_2 > N_1$, AP1 is a 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, the RSSI scheme has good performance while AP2 is less loaded, but when it becomes more loaded ($N_2 > N_1$) the RSSI scheme performs worse, because it selects the wrong AP. On the other hand, the load based AP selection scheme has inferior performance because it has only partial information about the load of AP1 due to its distance. The load based scheme in many cases chooses AP1 because it seems to be less busy, which is wrong when $N_2 < N_1$. Clearly, when $N_2 > N_1$, AP1 becomes the best choice, and the load based achieves good performance. While the RSSI scheme gives good results for $N_2 < N_1$ and the load based scheme for $N_2 > N_1$, our scheme gives good results for all values of N_2. To summarize, compared to the RSSI and the load based schemes, our scheme performs better in all situations that arise in this scenario, thanks to its ability of learning all the environmental aspects that affect performance.

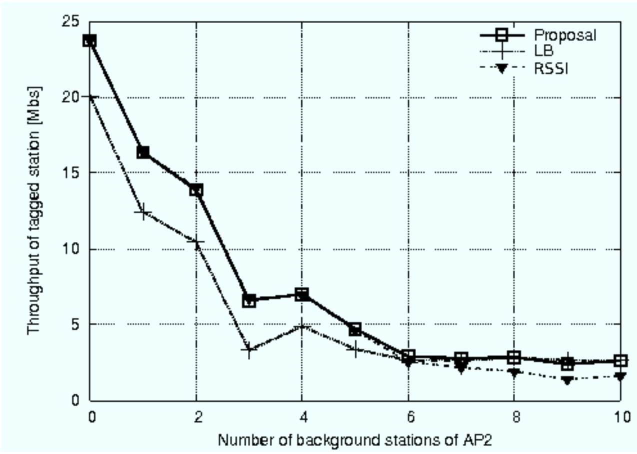

Fig. 5. Performance of different AP selection schemes in scenario with two AP, for different values of load of AP2, while load on AP1 is fixed as well as distance from tagged station to both APs

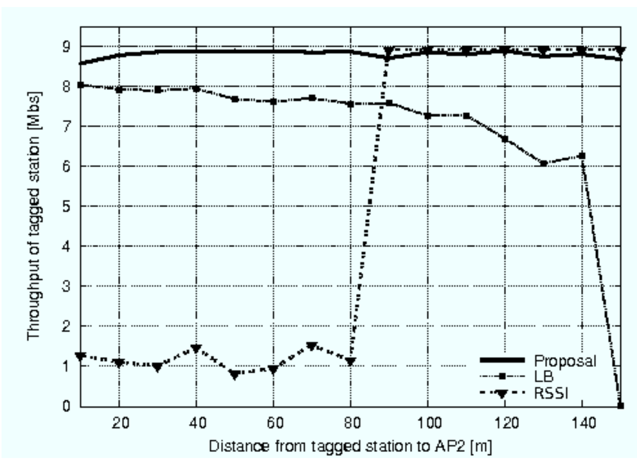

Fig. 6. Performance of different AP selection schemes in scenario with two AP, for different values of distance from tagged station to AP2, while distance to AP1 is fixed as well as load on both APs

2) Scenario with varying distance: The second group of experiments is characterized by fixed values N_1, N_2, d_1 and variable d_2. The settings for this scenario are summarized in table I. In Figure 6 we show the performance of our scheme, compared with the RSSI and the load based schemes. For $d_2 < d_1$ we see that the RSSI based scheme results in a wrong decision because it always connects to AP2, which is closer but much more loaded than AP1. Conversely, for $d_2 > d_1$ the RSSI scheme will perform better because it will always choose AP1, which is in this case closer to the tagged station. The load based scheme provides better performance than RSSI, but as the distance from AP2 increases it starts to choose more often AP2, since it considers it as less loaded. This is due to the fact that environmental information becomes more partial as the distance increases, as we explained in Section II.B. It can be noticed that the Neural Network based scheme has very good performance for all values of d_2 and performs better than the RSSI and the load based schemes.

IV. Conclusions

In this paper, we proposed a cognitive scheme based on learning. Our scheme is decentralized and uses Neural Network for the implementation of its learning capabilities. Using our scheme, the mobile devices is able to select the AP that is expected to yield the best throughput according to the past experienced performance. We described in detail the proposed cognitive solution, highlighting the aspects that make readily implementable in real mobile terminal. We carried out a performance evaluation by means of simulations and compared our cognitive solution to other schemes which are proposed in literature. The simulation results showed that our cognitive scheme is able to achieve significant improvements in throughput performance when compared with state of the art decentralized AP selection schemes in variety of 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] IEEE Std. 802.11-2007, “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications,” June 2007.

[2] L.-H. Yen and T.-T. Yeh, “SNMP-Based Approach to Load Distribution in IEEE 802.11 Networks,” in *Vehicular Technology Conference (VTC)*, May 2006.

[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] 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.

[5] V. Ghini, S. Ferretti, and F. Panzieri, “A strategy for best access point selection,” in *Proceedings of Wireless Days, 2010*, October 2010.

[6] Y. Fukuda, T. Abe, and Y. Oie, “Decentralized access point selection architecture for wireless LANs,” May 2004.

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

[8] “The ns-3 network simulator.” [Online]. Available: http://www.nsnam.org/

[9] N. Baldo, M. Requena, J. Nunez, M. Portoles, J. Nin, P. Dini, and J. Mangues, “Validation of the ns-3 IEEE 802.11 model using the EXTREME testbed,” in *Proceedings of SIMUTools Conference*, March 2010.

[10] E. Weingartner, H. vom Lehn, and K. Wehrle, “A Performance Comparison of Recent Network Simulators,” in *Proceedings of IEEE International Conference on Communications*, June 2009.

[11] “Pcap Trace Parser.” [Online]. Available: http://iptechwiki.cttc.es/Parser

[12] “Fast Artificial Neural Network Library (FANN).” [Online]. Available: http://leenissen.dk/fann/

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

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

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

[16] G. Holland, N. Vaidya, and P. Bahl, “A rate-adaptive MAC protocol for multi-Hop wireless networks,” in *Proceedings of Mobile Computing and Networking conference (MobiCom)*, 2001.