In building Localization Using Neural Networks

Uzair Ahmad, Andrey Gavrilov, Sungyoung Lee
{uzair, avg, sylee}@oslab.khu.ac.kr
Computer Engineering Department, Kyung Hee University,
South Korea

***Abstract-* Location Awareness is key capability of Context-Aware Ubiquitous environments. Received Signal Strength (RSS) based localization is increasingly popular choice especially for indoor scenarios after pervasive adoption of IEEE 802.11 Wireless LAN. Fundamental requirement of such localization systems is to estimate location from RSS. Multipath propagation effects make RSS to fluctuate in unpredictable manner, making location estimation unreliable. Moreover, in real life situations RSS values are not available at some locations making the problem more complex. We employ Modular Multi Layer Perceptron (MMLP) to effectively reduce the uncertainty in location estimation system. Our system provides robust location estimation with improved accuracy and caters for unavailable signals at estimation time.**

Key Words: Artificial Neural Networks, Location Aware Computing, Wireless LAN

1 INTRODUCTION

Location information is an integral and crucial component of many mobile computing applications [1] [2] [3] [4] [5] [12]. Pervasive adoption of IEEE802.11 (a, b, g) Wireless LAN (WiFi) has increased the potential of Location-Awareness technology to become a common service. Since signal strength measurements must be reported by the wireless network interface card as part of standard compliance, Positioning using Wireless LAN received signal strength (RSS) is a popular choice.

WiFi RSS based location awareness applications include, but are not limited to, a wide range of services to the end user like automatic call forwarding to user's location, robotic global localization, exploration and navigation tasks, Finder, Guiding and Escorting systems, first hop communication partners, liaison applications, location based advertisement and positioning of entities in large warehouses. We are developing Location awareness capability for ubiquitous computing middleware CAMUS [21]. Basic concept behind WiFi RSS based location awareness is that received signal strengths, of different Access Points (APs), follow certain patterns at a particular location. If these patterns can be captured and bound to that particular location, later when some device observes the same pattern location of that device can be estimated by processing previously captured patterns database. Process of capturing the RSS at every target location in a site is called 'calibration'. Fig 1 shows the methodology of calibration process.

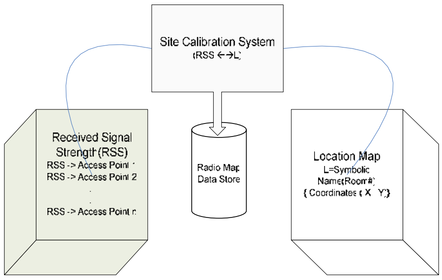

Fig 1: Site calibration methodology

Result of site calibration is radio map, a database that contains sample RSS values of all APs at all target locations. Once a Radio Map is built, it is used to develop a mapping function between target locations and respective RSS patterns. This function is later employed to estimate location of a device given RSS values. Indoor radio signal propagation follows a complex model due to multi-path effects, Non Line of Sight (NLOS) problem, human body absorption, neighboring devices and dynamic nature of environments. In this paper we present our experiments to develop Artificial Neural Network system for location estimation based on received signal strengths. Next section provides and overview of the previous work. In section 3, our Modular Multi Layer Perceptron approach is presented. Section 4 describes Design and Implementation details of the system. Results analysis is presented in section 5.

2 RELATED WORK

There have been several efforts to develop Location Aware system based on RSS. Bayesian classification and filtering [4] [13], Statistical learning theory [8], *K*-Nearest Neighbors [14] and Kalman Filtering [15] have been employed for this problem. Indoor wireless signal propagation is so complicated and elusive that it is still hard to achieve and maintain reasonable accuracy level for indoor location estimation systems based on RSS. Interacting with noisy, Non-stationary data that follows a complex process is very difficult using classical mathematical and engineering methods. Neural networks provide massive parallelism, fault tolerance and adaptation to circumstances. In dynamic indoor environments it is easier to let the neural network learn from examples. Only Battiti *et al* [16] have employed neural networks for this problem. They used feed forward back propagation network that takes RSS of different Wireless Access Points (AP) as input to the network. This model assumes that all the inputs are available at every location all the time. Practically, this approach has limited applicability because in real life scenario some AP may not be visible (not in range) at all the locations for all the time. We employ a modular approach that perfectly caters for this situation.

3 OUR APPROACH

Since all target locations are available, in addition to RSS values at each location, supervised learning is used to recognize signal patterns. The problem of constantly fluctuating RSS and even absence of wireless signal introduces very unreliable location estimations. Estimation reliability is directly affected by how good sample wireless signal data at target locations represent the real life situations. Therefore, we managed to collect a large number of RSS samples at each target location, contrary to the previous approaches, after a detailed site calibration. Particulars of site and its radio map are given in section 4.1. Multi Layer Perceptron feed forward back propagation neural networks has been employed by many researches for pattern recognition problem [6] [7]. But same approach is not sufficed to our problem due to unpredictable absence of signal in real life. We propose a modular approach to cope with uncertainty effectively. Details of our architecture are given in next section.

3.1 Modular Multi Layer Perceptron (MMLP)

A close observation of radio map gives important clues for using MMLP. Visibility of a signal allows filtering out possibility of unlikely locations and vice versa. Therefore for each set of available signal we employ a separate MLP neural network with best results. The overall architecture includes and one neural network at the beginning of estimation with binary inputs. Output of this neural network allows selection of next appropriate module. The criterion of selection is visibility of Access Points (AP) at a particular location. For our experimental setup this selection criteria is summarized in Table 1.

TABLE 1

Visibility Matrix

AP-1	AP-2	AP-3	Module
1	1	1	123
1	1	0	12
0	1	1	23

Three neural networks, with RSS of different APs on the input layer, are employed to estimate the exact location. Modular Multi Layer Perceptron architecture is shown in Fig 2. Our experiments were conducted with many different variants of MMLP architecture. Fig 2 is a particular instance of MMLP only to convey the basic idea.

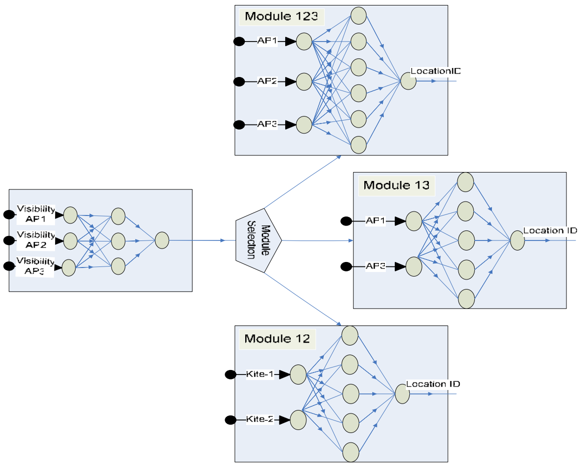

Fig 2: Modular Multi Layer Perceptron (MMLP) Architecture

4 DESIGN AND IMPLEMENTATION

We conducted experiments in 3rd floor of Engineering Building. Fig 3 shows the map, target locations and location of wireless Access Points. We divided all target locations into three regions. Corridor 1 is horizontal corridor with 10 target points. All point in corridor 1 region are given IDs from 11~20 from right to left. Corridor 2 is right vertical corridor with 6 target points. All points in corridor 2 region are numbered as 21~26. Similarly Corridor 3 is left vertical corridor with 6 target points. All points in corridor 3 region are numbered as 31~36. Two corner locations are termed as zero points as shown in Fig 3.

4.1 Calibration Phase

We collected 300 samples of RSS from all three Access Points at each location in calibration phase. Three IEEE802.11 (a, b, g) 3COM Access Points have been deployed in three corridors, as shown in Fig 3. We developed a device driver interface to capture the signal strength based on NDIS specification. NDIS protocol driver acts as a "relay" between an application and the NDIS miniport driver. A laptop equipped with integrated Intel wireless network interface card was used to build the radio map of the environment. IEEE 802.11 (a, b, g) specifies that signal strength measurement must be reported by the network interface card (NIC) as part of standard compliance [22]. The RSSI is measured in dBm and normal values for the RSSI value are between -10 and -100 [23]. Signal strengths recorded at each location are stored in a database called "Radio Map". Later this radio map is used to provide training samples for different neural network modules.

Fig 3: Location Map, Target Locations and location of Wireless Access Points

Graph shown in Fig 4 is made of a subset of the radio map, with location IDs on x-axis RSS values on y-axis. Signals from all three Access Point can be received on these points.

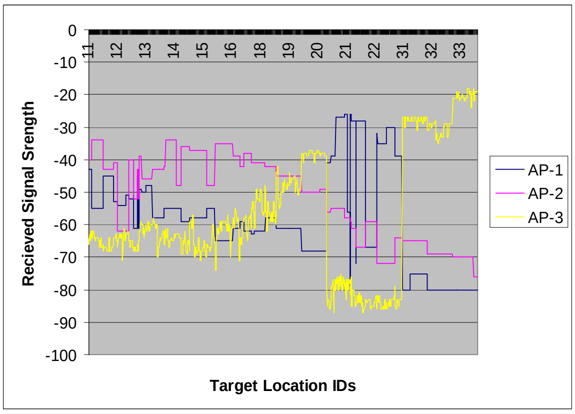

Fig 4: Points where all APs are accessible

Fig 5 graph shows some locations where signal of only two access points can be received.

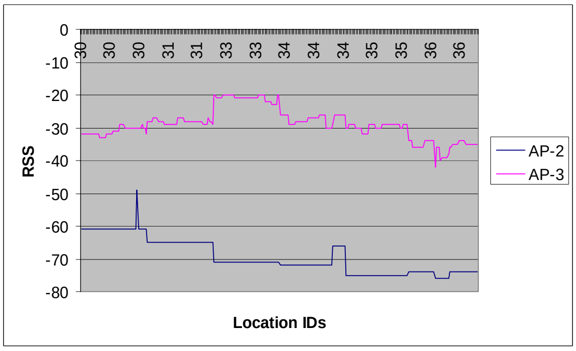

Fig 5: Points where only AP-2 and AP-3 are accessible.

Similarly figure 6 shows different set of access points accessible at different locations. Purpose of presenting radio map here is to emphasize the incompleteness and dynamic nature of RSS data at different locations. Device at two different locations can sometimes report same RSS readings, and can report very different readings while at the same location. This dilemma is main obstacle for getting absolute correct performance with the techniques mentioned in section 2.

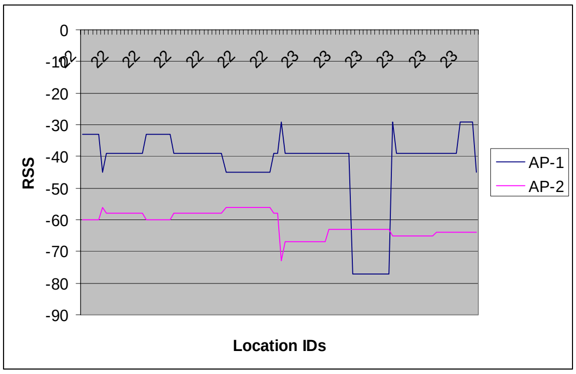

Fig 6: Points where only AP-1 and AP-2 are accessible.

4.2 Training Phase

Training phase is used to train different neural networks and analyze their comparative performance. Fig 7 shows system components that are involved in training phase.

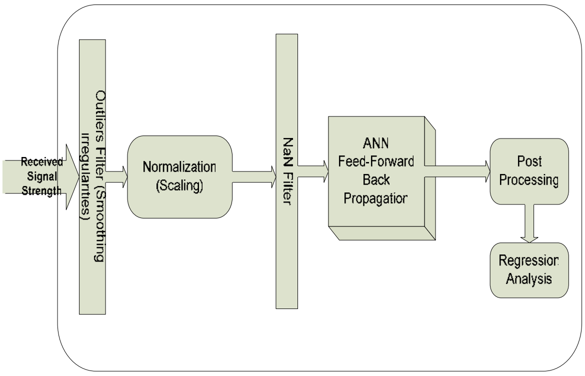

Fig 7: Neural Network Training System

Radio Map generated in Calibration phase is not used exactly. During calibration we observe certain RSS behaviors and apply statistical learning techniques to characterize signal strength properties at a particular location. Based on those characteristics, we implemented smoothing filters to remove outliers from RSS patterns. First component of training system is responsible for this task. Next component normalize the data and targets by scaling. After normalization some RSS values become too small to be effective in neural network learning. Such values are filtered by NaN (Not a Number) filter component. Then training sets are presented to neural network for learning the patterns of RSS and binding them with respective Location IDs. After learning is complete, data post processing component convert results back into un-normalized vectors. Regression analysis component is implemented to analyze the results.

We employed several configurations for finding the optimal location estimation accuracy.

TABLE 2
Training Performance of different Configurations

Algorithm	Performance	Goal	Epoc-hs	Structure	Transfer Func	
					Hidden Layer	Output Layer
RP[i]	0.01338	0.01	10000	381	Tan	Lin
SCG[ii]	0.01371	0.01	10000	381	Tan	Lin
CGB[iii]	0.01599	0.01	3000	381	Tan	Lin
LM[iv]	0.01328	0.01	1000	381	Tan	Lin
LM	0.01080	0.01	5000	381	Log	Lin
LM	0.00999	0.01	398	3881	Log	Lin
LM	0.00866	0.001	5000	3881	Log	Lin
LM	0.00866	0.001	3700	3881	Log	Tan

Table 2 contains our experimental neural networks used in training phase. Training properties in terms of goal, performance and required epochs are listed. Our experiments cover different configurations of feed forward back propagation networks. Back propagation is learning mechanism that allows a network to learn the relationships between inputs and outputs. It is a mathematical procedure that starts with the error at the output of a neural network and propagates this error backwards through the network to yield output error values for all neurons in the network. A common form of learning is "trial and error". A "trial" is the output of a system in response to particular stimuli. An "error" is the external reaction to the output of the system that is supplied to the system as some other kind of stimulus. A system capable of "trial and error" learning relies on receiving feedback that describes the nature and severity of mistakes. The system can use the error information to make corrections in the way it responds to that particular combination of stimuli in the future. Back-Propagation yields neuron error values throughout a neural network. Learning occurs when neuron input weights and bias values are adjusted in an attempt to reduce the output error for the same stimuli. Different other learning algorithms have been employed to train the networks in best possible way. All of these algorithms use the gradient of the performance function to determine how to adjust the weights to minimize performance. The gradient is determined using a technique called back-propagation, which involves performing computations backwards through the network. The back-propagation computation is derived using the chain rule of calculus and is described in Chapter 11 of [7]. Four training algorithms were chosen based upon literature review on supervised learning for pattern recognition with feed forward back propagation neural networks. Resilient back-propagation uses only the sign of the derivative to determine the direction of the weight update; the magnitude of the derivative has no effect on the weight update. The purpose of the resilient back-propagation training algorithm is to eliminate harmful effects of the magnitudes of the partial derivatives. A complete description of the resilient back-propagation algorithm is given in [6]. In the conjugate gradient algorithms a search is performed along conjugate directions, which produces generally faster convergence than steepest descent directions. Discussion on conjugate gradient algorithms and their application to neural networks is given in [7]. The scaled conjugate gradient algorithm (SCG), developed by [11] was designed to avoid the time-consuming line search.

---

[i] Resilient Propagation
[ii] Scaled Conjugate Gradient
[iii] CGB: Conjugant Gradient Powell/Beale Restarts
[iv] LM: Levenberg-Marquardt

For all conjugate gradient algorithms, the search direction will be periodically reset to the negative of the gradient. The standard reset point occurs when the number of iterations is equal to the number of network parameters (weights and biases), but there are other reset methods that can improve the efficiency of training. One such reset method was proposed by Powell [17], based on an earlier version proposed by Beale [10]. Levenberg-Marquardt (LM) algorithm appears to be the fastest method for training moderate-sized feed-forward neural networks (up to several hundred weights) [9]. The application of Levenberg-Marquardt to neural network training is described in [7]. In order to avoid over fitting problem of neural networks early stopping method was used. Mean Square Error (MSE) performance function was employed to measure the network errors. We performed our experiments using MATLAB neural network tool box [18]. As Table 2 suggests, Levenberg-Marquardt algorithm performed best in terms of faster pattern learning and goal achievement. After adding one more hidden layer to network structure, the performance goal 0.001 was achieved met with this algorithm.

4.3 Execution Phase

After training phase live data from the environment need to be tested with trained neural networks. In execution phase RSS captured on mobile device is presented to the input layer of neural network. After the number of accessible AP is determined, different preprocessing components are implemented to filter, scale and normalize data. Fig 8 shows all the components involved in execution phase.

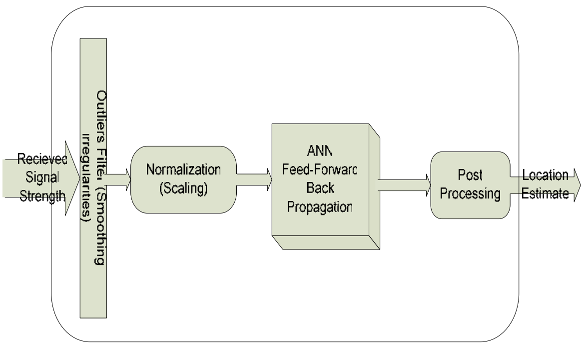

Fig 8: Execution of Location Estimation System

Outliers filter component is implemented to remove spikes from RSS data. Normalization component is responsible to scale the inputs in a given range. Once normalized, RSS readings are presented to the appropriate Neural Network module. Out put of neural network is post processed to get the Location ID estimate. In next section we shall present performance some results.

5 RESULTS ANALYSIS

Results are presented as estimation error in terms of meters. We employ Manhattan distance between estimated and actual location to represent error.

TABLE 3

Execution Performance of different Network configurations

Structure	Transfer Func		Training Algo	Error (Meter)		
	Hidden Layer	Output Layer		Max Error	Avg Error	Median Error
381	Tansig	Linear	CGF	1.9884	0.3501	0.143
381	Tansig	Linear	RP	1.8392	0.2863	0.1114
381	Tansig	Linear	SCG	1.5867	0.2740	0.0713
381	Tansig	Linear	LM	1.6263	0.2833	0.1001
381	Logsig	Linear	LM	1.8311	0.1724	0.008
3881	Logsig	Linear	LM	2.1667	0.1258	0
3881	Logsig	Tan	LM	2.1667	0.1258	0

Table 3 summarizes all the network configurations that we tested for one module (with complete inputs). 3881 architecture with Levenberg-Marquardt Algorithm training algorithm produced best average performance with 0.1258 meter error in estimation. But this network produced the maximum error of 2.1667 meters at the same time.

In order to analyze the performance of location estimation system, it is needed to employ a comprehensive model that can balance the performance measure among all aspects of accuracy. We applied a comprehensive model for evaluation of location estimation techniques. It covers the all performance aspects. This evaluation model provides both qualitative and quantitative insight into performance of location estimation system.

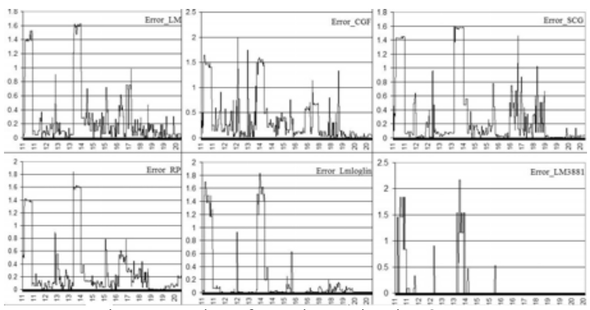

Fig 9: Execution of Location Estimation System

Fig 9 shows the error in estimation at every target location of the site. On x-axis of each graph, location ids are listed and on y-axis estimation error is plotted as a line graph. This shows location specific performance of different networks. It is obvious from these graphs that location estimation error is divided in two aspects i) over all error in the area ii) location specific error. Although 3881 structure produce highest error at two locations still it provides best accuracy overall aspect. This fact is obvious when a closer observation is made on to the

location specific results of each candidate neural network.

TABLE 4: Comparison with related work

Technique		AC[**]	AP[††]	SMP[‡‡]	Err[§§]
Neural Net (Trento)		624	3	200	1.75
Markov Chains (HIIT)		----	----	20	2.48
Triangulation (CMU)		60	3	3100	4.5
Bayesian Net (Rice)		30	9	100	1.5
Kernels (HKU)		100	3	100	3
KNN	RADAR	980	3	40	2.65
	UNM	160	5	20	3.15

Our system is implemented to cover 300 Square Meters ‘U’ shaped Area (see Fig 3) with 3 Access Points and 120 samples were taken for training at each location.

6 Conclusion And Future Directions

Employing IEEE 802.11(a, b, g) Wireless LAN as infrastructure for indoor Location Awareness is prudent choice due to its low cost and pervasive coverage. Since all Wireless Network Interface Cards have to report Received Signal Strength as dBm, it is very practical to implement localization capability based on RSS values at a particular location.

We employed a novel Modular Multilayer Perceptron architecture for Wireless LAN RSS based location estimation. This architecture provides robust mechanism for coping with unavailable information in real life situations. Experimental prototype was implemented for three corridors of Engineering Building 3rd floor. We evaluate our location estimation system performance with both overall and location specific measures. Table 4 summarized the results of previous researches. We considered four aspects of work that should be of interest to the readers.

Results show superior performance to previous approaches.

In future this system shall be extended to cover larger area including rooms and laboratories. In order to make this system available on demand for mobile end users, it is required to implement it as a software component. In future we plan to provide this system as a middleware service as explained in [19] [20]. This location service is part of Context Aware Middleware for Ubiquitous Computing (CAMUS) project.

7 References

[1] P. Bahl, *et al.* “A software system for locating mobile users: Design, evaluation, and lessons“, Microsoft Research, MSR-TR-2000-12, April 2000.

[2] P. Bahl, *et al.* ”RADAR: An in-building RF-based user location and tracking system.” In *IEEE INFOCOM 2000*, pages 775–784, March 2000

[3] Cynthia *et al,* “Challenges in Location-Aware Computing” Published by IEEE ComSoc 1536-1268/03/ 2003 IEEE

[4] Andrew *et al,* “Using Wireless Ethernet for Localization”, 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems

[5] Mike Hazas *et al,* "Location-Aware Computing Comes of Age," *Computer*, vol. 37, no. 2, pp. 95-97, February, 2004.

[6] Riedmiller, M., and H. Braun, "A direct adaptive method for faster backpropagation learning: The RPROP algorithm," Proceedings of the IEEE International Conference on Neural Networks, 1993.

[7] Hagan, M. T., H. B. Demuth, and M. H. Beale, Neural Network Design, Boston, MA: PWS Publishing, 1996.

[8] Roberto Battiti *et al*, “Statistical Learning Theory For Location Fingerprinting In Wireless Lans”, October 2002 Technical Report # DIT-02-0086 University of Trento Italy

[9] Hagan, M. T., and M. Menhaj, "Training feedforward networks with the Marquardt algorithm," IEEE Transactions on Neural Networks, vol. 5, no. 6, pp. 989-993, 1994.

[**] Area Covered in Square Meters

[††] Number of Access Points used

[‡‡] Number of Samples used per location

[§§] Accuracy measure in Euclidean distance error

[10] Beale, E. M. L., "A derivation of conjugate gradients," in F. A. Lootsma, ed., Numerical methods for nonlinear optimization, London: Academic Press, 1972.
[11], "A scaled conjugate gradient algorithm for fast supervised learning," Neural Networks, vol. 6, pp. 525-533, 1993.
[12] R. Want and B. Schilit, "Expanding the horizons of location-aware computing", *IEEE Computer*, 34(8):31–34, August 2001.
[13] Dieter *et al,"* Bayesian Filters for Location Estimation", *IEEE CS and IEEE ComSoc* 1536-1268/03
[14] K. Pehlavan *et al,* "Indoor Geolocation Science and Technology", IEEE Communications Magazine, 2002
[15] Guvenc *et al*, "Enhancements to RSS Based Indoor Tracking Systems Using Kalman Filters," GSPx & International Signal Processing Conference, Dallas, TX, March 31-April 3, 2003.
[16] R. Battiti *et al*, "Neural Network Model for intelligent networks: deriving the Location from signal patterns, The First Annual Symposium on Autonomous Intelligent Networks and Systems 2002
[17] Powell, M. J. D., "Restart procedures for the conjugate gradient method," Mathematical Programming, vol. 12, pp. 241-254, 1977.
[18] Neural Network Toolbox for MATLAB www.mathworks.com/products/neuralnet/
[19] Uzma Nasir *et al*, "On Building a Reflective Middleware Service for Location-Awareness", The 11 IEEE International Conference on Embedded and Real-Time Computing Systems and Application (RTCSA), Hong Kong, 17-19 August, 2005
[20]Mahrin Iqbal *et al*, "Reflective Middleware for Location-Aware Application Adaptation. ICCSA (2) 2005: 1045-1054
[21] Saad *et al,* "A Distributed Middleware Solution for Context Awareness in Ubiquitous Systems", 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'05) pp. 451-454
[22] Steve Pope, "Issues Related to RSSI Measurement", IEEE 802.11-02/520r0
[23] "IEEE 802.11 Network Adapter Design Guidelines for Windows XP", Microsoft, May 22, 2003 http://www.microsoft.com/whdc/ device/