Macrothink Institute™

Network Protocols and Algorithms
ISSN 1943-3581
2014, Vol. 6, No. 4

RSSI Prediction in WiFi Considering Realistic Heterogeneous Restrictions

Alvaro Suarez[1,3], J. Aurelio Santana[2,3], Elsa Macías[1,3], Vicente Mena[2,3], J. Miguel Canino[2] and Domingo Marrero[1,3]

[1] Departamento de Ingeniería Telemática, Universidad de Las Palmas de Gran Canaria, Spain

{ alvaro.suarez, elsa.macias, domingo.marrero}@ulpgc.es

[2] Departamento de Señales y Comunicaciones, Universidad de Las Palmas de Gran Canaria, Spain

{jose.santana, vicenteefigenio.mena}@ulpgc.es

jcanino@ dsc.ulpgc.es

[3] Grupo de Arquitectura y Concurrencia, Universidad de Las Palmas de Gran Canaria, Spain

Received: August 4, 2014 Accepted: October 2 2014 Published: December 27, 2014

DOI: 10.5296/npa.v6i4.6066 URL: http://dx.doi.org/10.5296/npa.v6i4.6066

Abstract

Received Signal Strength Indicator is an indication of power level being received by a wireless device. This parameter for Wireless Fidelity has never been standardized, so any manufacturer implements it measuring as he knows better doing. This is a problem because different scenarios can be found based in several elements like: the particular Wireless Fidelity Network Interface implementation, the driver, the operating system and software monitoring implementation. This makes the utilization of the Received Signal Strength Indicator to implement higher level functions in wireless networks (like roaming management or localization of terminals) be a headache in realistic scenarios. This heterogeneity of devices, drivers… makes that the application of the Received Signal Strength Indicator be not general. In this paper we present the comparison of our new model of Received Signal Strength Indicator prediction in several realistic and heterogeneous scenarios. The experimental results indicate that we obtain in general, better results than the other models.

Keywords: Received Signal Strength Indicator, Wireless Fidelity, Prediction, Regression Analysis.

1. Introduction

Wireless Fidelity (*WiFi*) is a communication technology that is used in our days in all around the world in several kinds of environments. Its applicability in a lot of scenarios is very high. A WiFi device (emitting device) uses an antenna to propagate a wireless electromagnetic signal in all the space directions or in a sectorial area of that space. Another WiFi device (receiving device) in the coverage area of the first one can receive the propagated signal. The main problem here is that the original propagated signal will not be perceived by the receiving device in its original transmission state. That is, the signal when travels in the space will be degraded. This degradation depends on the distance between the devices basically but other factors also affect that degradation.

The *Received Signal Strength Indicator* (*RSSI*) is a read only property that serves as an indication of the power level being received by a receiving device's antenna at a particular distance from the emitting device. Mainly, RSSI values increase or decrease as the relative distance between the emitting and receiving devices decreases or increases. But others factors influence these values. RSS is related to *Signal-to-Interference-plus-Noise Ratio* (*SINR*) [1] that includes the RSS and the interference caused by other nodes plus noise. That is, RSSI is sensitive to the channel noise, interference, reflections and attenuations, and it suffers from the antenna variability. Therefore, RSSI values exhibit high variability in space and time on real life WiFi networks in diverse environments. Moreover, one of the main reasons to use WiFi technology is that devices can move freely inside the coverage area. That is, if the moving device is receiving low values of RSSI this will indicate (theoretically), that it is at a long distance from the emitting device. This means that it probably could go out of the coverage area in a short period of time. This means that the prediction of RSSI values is very useful in order to prevent faults of communications among WiFi devices. For this reason it is very important to predict efficiently the future RSSI values taking into account the recent historic received values. In general prediction of RSSI can be used in other domains: distance estimator [2] and device localization mechanism [3], adaptive data and power transmission [4], link level quality estimation [5], terminals connectivity indication to *Access Point* (*AP*), service disruption anticipation in the event of predicted link degradation or failure to support a handover process [6].

Recently models have been proposed that predict RSSI values to anticipate these future values to particular applications. The work presented in [6] shows a simple protocol of handover in presence of multimedia information using a filter and predictor based on RSSI estimations. The work in [7] proposes the modelling of wireless mesh networks using a modified Ornstein–Uhlenbeck jump diffusion process [8] [9] [10]

[11]. A predictor for applications that needs to anticipate RSSI values hundreds of milliseconds in advance is presented in [12]. It has the advantage to be location-independent. The proposal is based on a regression model of smoothed recent past measurements of the RSSI. The work in [13] operates over a window of N raw RSSI samples, fitting a 8-degree polynomial over those samples, and predicting the following RSSI value simply evaluating that polynomial with the last known raw RSSI sample. The above works do not consider the complexity of the realistic WiFi networks, the diversity of devices in the Market, techniques for measuring the RSSI, complexity of drivers of the Wireless Network Interface Cards (*WNIC*)… The main contribution of this paper is a novel model to predict RSSI values. We demonstrate that there are realistic scenarios where our predictions are closed to real measures and behave better than other models. In general, for all the revised models it is difficult a practical implementation that behaves properly under a wide variety of scenarios.

The rest of the paper is organized as follows: in Section 2 we present the difficulties of predicting the values of RSSI in the actual Market devices and related work of other predicting models. In Section 3 we present our model to predict the RSSI values analyzing its mathematical basis. In Section 4 we present the experimental considering several realistic and practical cases and finally we present the conclusion.

2. Received Signal Strength Indicator in WiFi

RSSI is commonly available in *Commercial Off-The-Shelf* (*COTS*) wireless card hardware. All COTS deliver a numeric value defined for the *Electrical and Electronics Engineers* (*IEEE*) 802.11 standard [14] to represent the strength of received *Radio Frequency* (*RF*) signal. The RSSI is read from the internal register of transceiver or physical controller using the corresponding driver. Dynamically the values are updated from the electric signals (physical level) contained in Medium Access Control's frames (denominated beacons) in the associated channel. That means that if the beacons are transmitted every 100 ms, the receiving device will detect until 10 RSSI's values in a second (although it is a very difficult to detect all of them).

In general the values are delivered to the user (programmer or simply end user) in *Decibel for milliwatt* (*dBm*), and they are presented as a percentage of the signal strength (related to the signal noise relation) or as an absolute value. A first challenge we find in actual COTS is that not all the WNIC provide the RSSI values in the same scale [15]. This is because the IEEE 802.11 does not force the usage of a particular format and range to deliver RSSI values. It only specifies that RSSI is an optional parameter to be measured. For that reason each vendor implements its hardware and drivers freely choosing the range of RSSI values and units (milliwatts (mw) or dBm). And what is worst, the vendors transform the RSSI dBm values to adjust them in a linear or quadratic form. Typical ranges of RSSI values are [0..60], [0..100] or [0..127]. The problem of this is confusion: when values are read from a particular device the scale and the way the driver deliver them is not known. For this reason it is difficult to implement applications based on RSSI.

The element in charge to deliver the above values to the user is the driver. A second challenge we find is the driver calculates an average of the values of received signal level. For that reason the design and implementation of the driver is critical. Two realistic Linux utilities [16] probe this: a) *iwconfig* receives averaged RSSI values from the driver and informs the user about this average, but b) *iwlist* obtains a sample of each WiFi channel after a fast scanning of all the channels (the number of AP detected change along the time). That is, *iwlist* only detects the physical header of the beacons sent by the AP (in this header there is the raw value of RSSI). This means that the measure of RSSI values from the WNIC is not simple as only capture them. It is important to know how those values have been measured by the COTS and the driver.

The above two challenges are illustrated in Fig. 1. We measured RSSI using three different devices with different operating systems and we did the measures exactly in the same place with the same environment restrictions. In the Fig. 1.a and 1.b the range of values is delivered in dBm (negative values), in the Fig. 1.c the driver delivers only positive values. The average of values delivered by the device in Fig. 1.a and 1.b is low while in Fig. 1.c is very high (continuous constant value of RSSI can be appreciated). This shows that the influence of the measured values of RSSI is high due to the actions done by the driver and the monitoring RSSI application (iwconfig, iwlist, tcpdump or wireshark, native application of Microsoft Windows, iOS or Android…).

WiFi technology uses two physical Industrial Science and Medical bands at 2.4 and 5 GHz. The 2.4 GHz band is very crowded (a lot of devices use them in our days). This means that RSSI is sensitive to the channel noise, interference, reflections and attenuations, and it suffers from the antenna variability. Therefore, RSSI values exhibit high variability in space and time on real life networks in diverse environments. Particularly ElectroMagnetic Interference (EMI, or radio-frequency interference) affects 2.4 GHz band. Powered AP and recent IEEE 802.11n [17] and IEEE 802.11ac use different non overlapped channels to associate the WiFi terminals for reducing EMI. But this effect is not radically eliminated due to adjacent channels can also interfere. Obstacles and boundary restrictions (especially indoors) provoke reflections and multipath fading. These effects can be reduced using *Multiple Input Multiple Output* (*MIMO*) antennas but not eliminated at all. The above versions of IEEE 802.11 can use this technology but not previous 802.11a/b/g [18] [19] [20]. Finally it is shown that atmospheric restrictions (especially outdoors) [21] can also produce undesired variations of perceived RSSI. A third challenge is to predict the RSSI values in the presence of these restrictions that provoke a random variability of the RSSI. To show this we have done two different experiments. First we arranged a static AP and a WiFi terminal. Then we measured the received signal level from a total amount of 3330 beacons from the AP in the terminal. We isolated the effect of the operating system making the measures using Monitor Mode of WiFi interface and the *RadioTap* [22] module to detect the received signal level. Traditional radio propagation models state that received signal and consequently the RSSI varies with the distance.

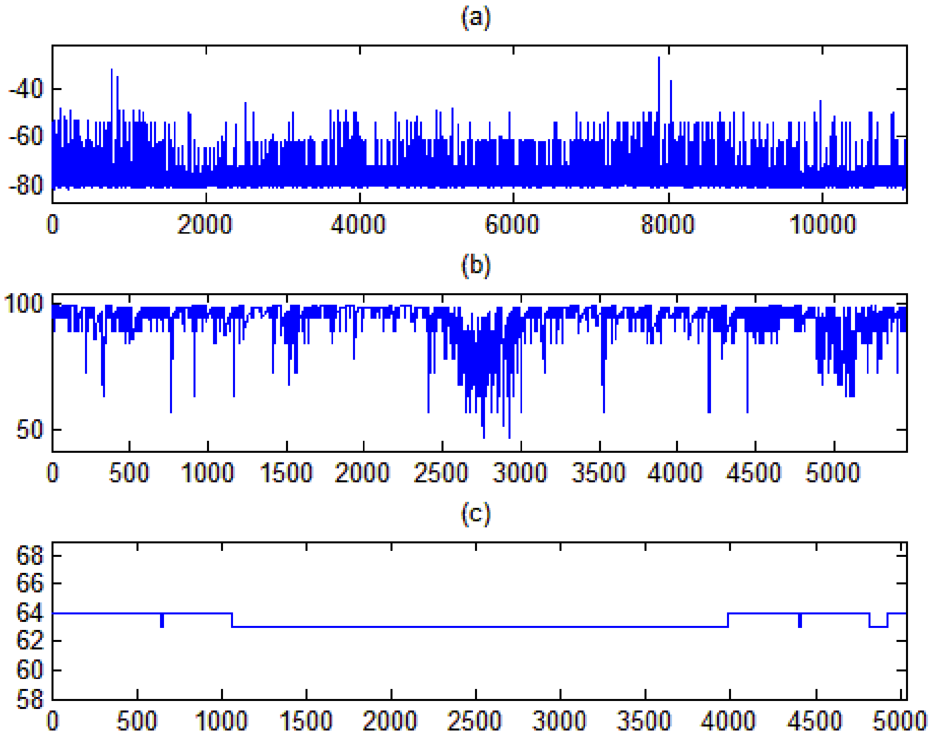

Figure 1. Different measures of RSSI in different devices and operating systems. a) iwconfig under Linux laptop, b) NetStumbler under Windows XP latptop, c) iwconfig under rooted Android eBook

That is, if we arrange the terminal at 50 m to the AP then the received signal level or the RSSI will must be constant for all the 3300 beacons. In contrast in Fig. 2 we show the signal variation is between -75 and -60 approximately and it varies randomly between these values. That is, the variation is not always the same for different measures campaigns. But it is true that the mean of the most of these values is a constant. That is, the most of the values can be adjusted to a straight line. The rest of the values (those that are too far from the mean, that we name jump values) must not be considered to do this adjustment. Second we moved the WiFi terminal leaving the AP static. The trajectory of the terminal was rectilinear and going away the AP (just to find a zone of no coverage). We moved the terminal meter by meter. In each stop we spent 30 s and measured a total amount of 1500 beacons. We moved till 30 m away the AP approximately. The propagation models state that the RSSI varies in a quadratic form depending on the square of the distance between the AP and the terminal. In contrast in Fig. 3 we show that again a random variation of RSSI was found. But it is true that we can adjust these values to a quadratic curve using the most of the RSSI values and disregarding the jump values.

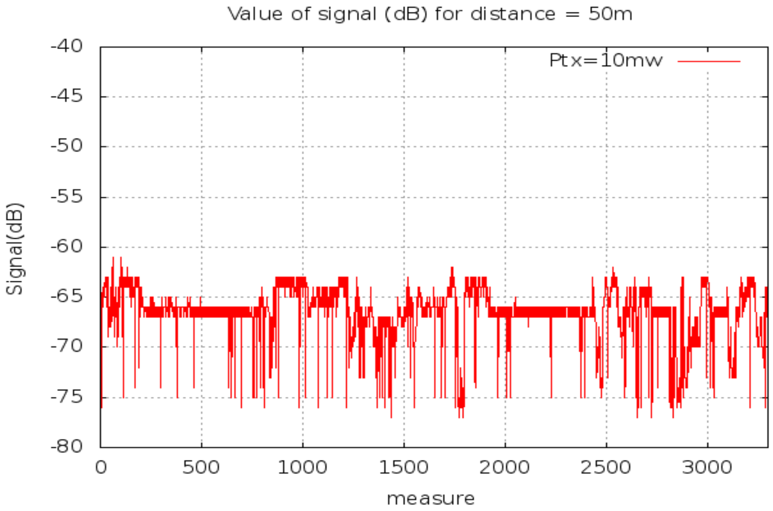

Figure 2. We measured the received signal level directly from beacon with a static WiFi AP and terminal for observing its random variation.

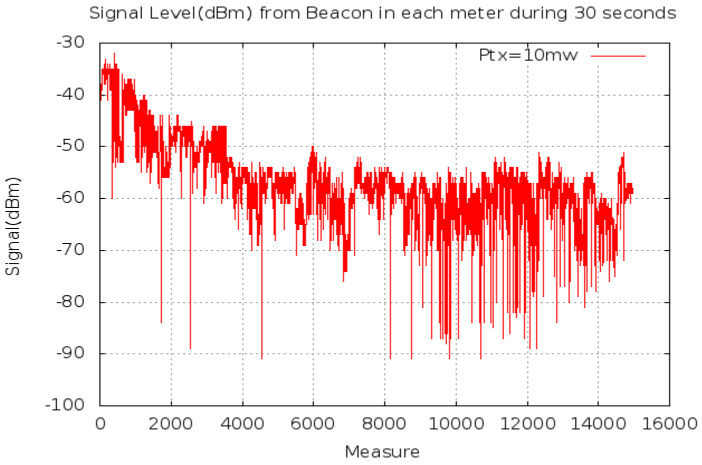

Figure 3. We measured the received signal level with a static WiFi AP and a moving terminal away the AP for observing its random variation.

As we have shown RSSI prediction models must take into account the random variation nature of WiFi RSSI. This prediction in realistic scenarios is very difficult because it is not trivial to eliminate the jump values.

The challenge of predicting RSSI values considering the jump values is very important due to in some cases these jump values could appear during a long time. In this case, the traditional (theoretical) propagation models fail and it is unlikely that a traditional RSSI predicting model that only adjust non jump values will work fine. The importance of this is that undesired disruption of service can appear while the predicting model does not warn about it with enough anticipation (as shown in measures between 10000 and 12000 approximately in Fig. 3). Localization and roaming is also affected by this likely inappropriate behavior of RSSI predicting models.

3. Related Work

Recently several RSSI predicting models have been presented. We now analyze them in order to fix their behavior in front of the three above challenges we have identified.

In [13] the authors work with the raw signal, trying to fit an 8-degree polynomial to an N raw RSSI sample window, and predicting the following sample from that polynomial.

In Long and Sikdar model [12] is presented a method where a linear regression model is used to fit previously smoothed RSSI data. They use variable size windows that consider the last read RSSI values and then predict the next value of RSSI. Window size is incremented after each prediction, unless forecasting error exceeds some threshold, in which case window size is reset to initial size.

In [7] is presented a method that also uses linear regression to fit smoothed RSSI data, in this case, to a modified mean reverting diffusion process called Ornstein-Uhlenbeck jump diffusion process [8][9][10][11]. It considers the selective elimination of jump values for predicting the next value of RSSI. But this method have been probed for only one kind of devices that include Intel © traces. Moreover they do not take into account interferences and complex movements of devices.

The two last resulting methods make a smooth of the original raw signal. In this way they treat to eliminate the effects of fading (jump values of RSSI). The inconvenience of this is that they do not take into account that the driver could do this previously, making duplicate actions.

Let us center our attention in the elimination of jump values observing the behavior of these two methods comparatively (Fig. 4) using a flow chart block diagram. Both charts share three blocks in which they implement different actions:

- *Read RSSI* block: They both consider a particular size of RSSI sample to be used for predicting. In [12] a dynamic windowing mechanism is used to adapt the abrupt variations in the Physical level WiFi signal. They start with a default window size and calculate it size iteratively while minimize the error ($e = yi+yi,$) between the real signal level (yi) and the predicted one (yi). Frequent jump values (sudden values) will cause the window size be reduced. But subsequent correct predictions will cause the window size be increased. To decide the increase or decrease of the window size the *e* is compared with a threshold. Also a maximum and minimum size of the window is defined. On the contrary in [7] is used a fixed window size.
- *Smooth RSSI* block: They implement a particular smoothing procedure of previous RSSI values in order to remove data variations due to small scale fading and measurement noise. In [12] was used the *triweight* kernel function that minimized the *Mean Integrated Squared Error* (*MISE*). In [7] was used uniform smoothing process to reduce the fluctuation in the raw signal values and helps to convert the time series data into a data set with fewer fluctuations.
- *Predict Future Signal Strength* block: They consider particular parameters required by the mathematical model that carry out the prediction. In [12] was used a linear regression model to predict the signal attenuation trend caused by path loss and shadowing. In [7] was also used a linear regression model but eliminating the jump values of RSSI. This is because they used a fixed window size.

Up to our knowledge the above methods do not consider the influence of heterogeneous drivers and operating systems nor monitoring applications for measuring the RSSI values.

4. The Proposed Prediction Model

The algorithm we designed intends to predict RSSI signal trend as accurate as possible using previous signal samples. We expect for this method to have a good behaviour in diverse environments and low computational complexity. We will obtain the future RSSI samples through a linear regression model like $\hat{X}_{N+p} = A \cdot X_N + B$, where the predicted value $\hat{X}_{N+p}$ corresponds with a modified mean-reverting diffusion process, called Ornstein-Uhlenbeck ([8] - [11]) process, which is a stochastic process that describes the velocity of a massive Brownian particle under the influence of friction.

Over time, the process tends to drift towards its long-term mean. Coefficients A y B are related to Ornstein-Uhlenbeck diffusion process as we will develop below. As we introduced above, other authors like [7] and [12] use similar approaches.

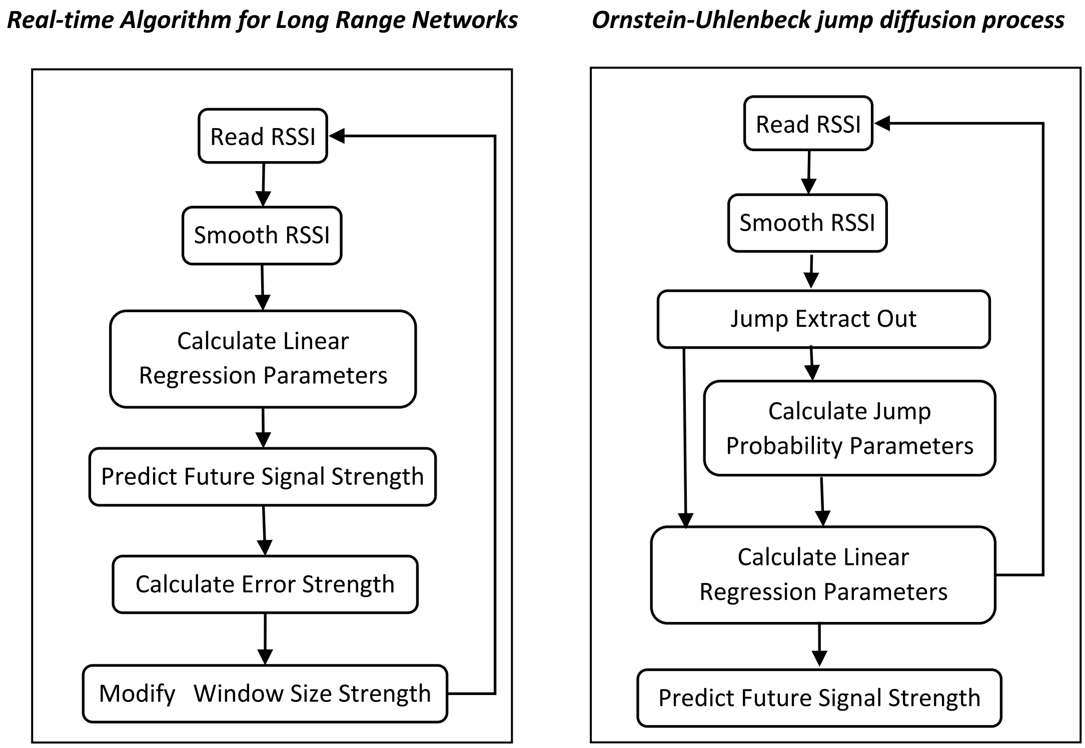
Real-time Algorithm for Long Range Networks
Read RSSI
Smooth RSSI
Calculate Linear Regression Parameters
Predict Future Signal Strength
Calculate Error Strength
Modify Window Size Strength
Ornstein-Uhlenbeck jump diffusion process
Read RSSI
Smooth RSSI
Jump Extract Out
Calculate Jump Probability Parameters
Calculate Linear Regression Parameters
Predict Future Signal Strength

Figure 4. Flow Charts for comparing Real-time Algorithm for Long Range Networks and Ornstein-Uhlenbeck jump diffusion process models.

Long and Sikdar model [12], given a N known Strength Signal sample set $X_i, i \in [1, N]$, predicts the following p values through a simple linear equation: $\hat{X}_{N+p} = a \cdot p + b$, where the coefficients are calculated as $a = \frac{V_{xy} - \bar{x} \cdot \bar{y}}{V_x^2 - N \cdot \bar{x}^2}$, $b = \bar{y} - b \cdot \bar{x}$, being V_{xy} the cross-covariance of X and Y; V_x^2 is the variance of X; and finally $\bar{x}$ and $\bar{y}$ are the means of X and Y respectively. Initially N has some N_{start} value which will be increased when the system acquires next RSSI value X_{N+p}. Obviously, not all the samples of RSSI will fit the regression curve accurately. When the predicted value $\hat{X}_{N+p}$ differs from the real RSSI sample X_{N+p} more than a threshold error e, N is reset again to N_{start} value. This strategy permits the algorithm to adapt to local signal trend, though it lacks in accuracy when N is reset. Unlike Long and Sikdar algorithm, our approach maintains a fix size window of known RSSI samples though it allows some level of variability from the general trend of the signal. In the other hand, regression is applied over RSSI amplitude values X_N, rather than over the prediction index p.

In [7] Ornstein-Uhlenbeck process is also used to explain RSSI signal tendency, and they called their method the *Modified Ornstein-Uhlenbeck jump diffusion process* (*MOUjd*), and its behaviour is described by (1).

$$dX_t = \kappa \cdot (\theta - X_t) + \sigma \cdot dW_t + \log J_t \cdot dN_t, \quad dW_t \sim N(0, dt) \quad (1)$$

Where $dW_t \sim N(0,dt)$ is a Wiener process, $\kappa > 0$ is the mean reversion rate, θ is the mean and $\sigma > 0$ is the volatility. The process dN_t is a Poisson process with parameter λ such that dN_t is 1 with probability λdt and 1- λdt otherwise. This latter Poisson term represents jumps in the general diffusion trend RSSI is supposed to follow (remaining terms in (1)). The random variable $J_t > 0$ is the jump amplitude with log $J_t \sim N(\mu_J, \sigma_J^2)$. And dW_t, dN_t and J_t are mutually independent. We can consider jumps to play a similar role as variable window size N in [12]. They give flexibility to both algorithms in order to permit certain deviation of some RSSI samples from the general signal trend. The solution of stochastic differential (1) is showed in (2).

$$X_{N+p} = X_N \cdot e^{-\kappa \cdot p \cdot \Delta t} + \left(\theta + \lambda \cdot \frac{\mu_J}{\kappa}\right) \cdot \left(1 - e^{-\kappa \cdot p \cdot \Delta t}\right) + \\ \sigma \sqrt{\frac{1 - e^{-\kappa \cdot p \cdot \Delta t}}{2 \cdot \kappa}} \cdot Z_1 + \sqrt{\frac{\lambda}{2 \cdot \kappa} \cdot \left(\mu_J^2 + \sigma_J^2\right)} \cdot Z_2, \quad Z_1, Z_2 \sim N(0,1) \quad (2)$$

As in (1) two terms were distinguished, the general signal trend and the jumps term, the same must be considered in order to evaluate (2). Firstly jumps must be identified for variables λ, μ_J and σ_J^2 be calculated. And, once jumps are extracted out, remaining variables can be estimated through regression.

Jumps are contiguous RSSI sample differences X_{i+1}-X_i which exceed some threshold. Empirical rule is often applied to classify values, stating that 99.7% of the values lie within exceeds three standard deviation of the mean. It can be showed in (3), simply by evaluating normal distribution between μ-$n \cdot \sigma$ and μ+$n \cdot \sigma$

$$\begin{aligned}
&\Phi_{\mu,\sigma^2}(x)=\frac{1}{\sigma\cdot\sqrt{2\cdot\pi}}\cdot\int_{-\infty}^{x} e^{-\frac{(u-\mu)^2}{2\cdot\sigma^2}}\cdot du\\
&\Phi_{\mu,\sigma^2}(\mu+n\cdot\sigma)-\Phi_{\mu,\sigma^2}(\mu-n\cdot\sigma)=\Phi_{0,1}(n)-\Phi_{0,1}(-n)=\\
&=\Phi(n)-\Phi(-n)=\Phi(n)-(1-\Phi(-n))=\\
&=\Phi(n)-(1-\Phi(n))=2\cdot\Phi(n)-1=erf\left(n/\sqrt{2}\right)=\begin{cases}0.682689492137, & \mathrm{n}=1\\ 0.954499736104, & \mathrm{n}=2\\ 0.997300203937, & \mathrm{n}=3\end{cases}
\end{aligned}\tag{3}$$

Thus, in every fix size *N* samples window, jumps are extracted out from RSSI differences which exceed more than three standard deviations of the mean. Mean and standard deviation are calculated again among the remaining RSSI differences, until no other jumps are identified. Final frequency λ, mean μ_J, and variance σ_J^2 of jumps will become part of prediction (2).

Once isolated general signal trend from jumps, regression coefficients can be calculated as described in (4.a)

$$X_{i+1}=a\cdot X_i+b+\varepsilon_i,\quad \varepsilon_i\sim N\left(0,\sigma_\varepsilon^2\right)\tag{4.a}$$

This linear equation must be solved through a regression analysis as indicated in (4.b), (4.c) and (4.d). Note that *a* coefficient, according to (2), must be an exponential whos exponent is always negative. Therefore its value has to be limited to *a*∈(0,1), although previous a_0 value exceeds this interval, adjusting *a* to be a minimum ε distance over 0 or down 1, according to which limit a_0 has crossed.

$$\begin{aligned}
S_x&=\sum_{i=1}^{N}X_{i-1}\\
S_y&=\sum_{i=1}^{N}X_i\\
S_{xx}&=\sum_{i=1}^{N}X_{i-1}^2\\
S_{yy}&=\sum_{i=1}^{N}X_i^2\\
S_{xy}&=\sum_{i=1}^{N}X_{i-1}X_i
\end{aligned}\tag{4.b}$$

$$a_0=\frac{N\cdot S_{xy}-S_x\cdot S_y}{N\cdot S_{xx}-S_x\cdot S_x}\tag{4.c}$$

$$\begin{aligned}
a&=\begin{cases}a_0, & 0<a_0<1\\ \varepsilon, & a_0\le 0\\ 1-\varepsilon, & a_0\ge 1\end{cases}\\
b&=\frac{S_y-a\cdot S_x}{N}
\end{aligned}\tag{4.d}$$

Equation (5) will give the last parameters to evaluate prediction (2) as suggested by [7].

$$\sigma_\varepsilon^2 = \sqrt{\frac{N \cdot S_{yy} - S_y^2 - a \cdot \left(N \cdot S_{xy} - S_x \cdot S_y\right)}{N \cdot (N-2)}} \qquad \theta = \frac{b}{1-a}, \quad \kappa = \frac{1-a}{\Delta t}, \quad \sigma = \frac{\sigma_\varepsilon}{\sqrt{\Delta t}} \qquad (5)$$

Our model, showed in (6), propose a simplified version of (2) to forecast which p RSSI values will be retrieved after a window of N known RSSI samples. We eliminated all the stochastic stuff in original formula, so we called it *Deterministic Ornstein-Uhlenbeck jump diffusion* (*DOUjd*) prediction method.

$$X_{N+p} = X_N \cdot e^{-\kappa \cdot p \cdot \Delta t} + \left(\theta + \lambda \cdot \frac{\mu_J}{\kappa}\right) \cdot \left(1 - e^{-\kappa \cdot p \cdot \Delta t}\right) \qquad (6)$$

Equation (6) avoids random terms but reaches a proper approximation as we show in Table 2 (next Section). Furthermore, it can be easily optimized and predicted values can be obtained with a expression almost so simple as [12] prediction formula, as we can see in (7).

$$X_{N+p} = X_N \cdot a^p + \gamma \cdot \left(1 - a^p\right), \quad a = e^{-\kappa \cdot \Delta t}, \quad \gamma = \theta + \lambda \cdot \frac{\mu_J}{\kappa} \qquad (7)$$

5. Experimental Results

In this Section we present experimental results for the above prediction models in realistic scenarios taking into account heterogeneous restrictions.

First of all we present the physical scenarios and the characteristics of WiFi heterogeneous devices. The measurements were done in our University (different buildings of the Telecommunication Faculty composed by laboratories and teachers' offices). We used different AP and WiFi terminals. For each terminal we used different operating systems and RSSI monitoring software. We defined 8 different scenarios (Table 1). In scenarios 1, 2, 3 and 4 signal strength measurement was done using *Linksys WRT54G/GL/GS* working with transmitted powers, 10 mW, 50 mW and 100 mW respectively. These measurements were done in a long corridor with offices on the sides. While we were doing the measurements people walked in the corridor uncontrollably; moreover the doors of offices were opened and closed uncontrollably. The rest measurements were done in a laboratory with *FON2200 Router* (63mW). This laboratory is a square room with several hardware equipments, tables and free space for walking.

In order to consider a realistic behavior of WiFi terminals we moved them at 1 minute per second approximately. The sampling interval was 0.25 seconds. For 2.4 GHz band (channel 6 in IEEE 802.11g) the radio wavelength (λ) was 0.1231 m. Thus, for example, predicting the signal strength sample 2 index steps ahead corresponds to 0.5 s (1.11 m ahead, which is about 10 λ).

Table 1: Heterogeneous and realistic restrictions of the scenarios

Scenarios	*AP*	*WiFi Terminal*	*SO/Software*	*Speed*
1	LinkSys WRT54G/GL/GS 10mW,	Laptop Acer Aspire ONE AO752	Linux Ubuntu/ tcpdump	Get away from 1m/min
2	LinkSys WRT54G/GL/GS 10mW,	Laptop Acer Aspire ONE AO752	Linux Ubuntu/ tcpdump	Static
3	LinkSys WRT54G/GL/GS 50mW	Laptop Acer Aspire ONE AO752	Linux Ubuntu/ tcpdump	Static
4	LinkSys WRT54G/GL/GS 100mW	Laptop Acer Aspire ONE AO752	Linux Ubuntu/ tcpdump	Static
5	FON2200 Router 63mW	Laptop Acer Aspire 1692WLMi	Linux	Approach 1m/min
6	FON2200 Router 63mW	eBook	Android	Static
7	FON2200 Router 63mW	Laptop Acer Aspire 1692WLMi	Linux	Static
8	FON2200 Router 63mW	Laptop Acer Aspire 1692WLMi	Windows /Netstumbler	Static

We analize and compare the results of applying Long and Sikdar, MOUjd and DOUjd RSSI prediction models. These models were tested in the 8 scenarioes described in Table 1. We fixed the prediction of *p* signal values, given a windows of *N* RSSI data samples. We have limited the study to *p*=1, i.e., we forecast one only signal value in all the cases. Data samples were acquired with a sampling time of (Δt) equals *to* 1 s. Our objective was to determine which method is more flexible to adapt to any environment (heterogeneous and realistic restrictions).

In order to compare these models we used the *Normalized Mean Square Error* (*NMSE*) to indicate the quality of forecast approximation $\hat{X}_i$ to real RSSI data sample X_i. Equation (8) describes NMSE expression we used, which is the equivalent to which was applied in [12] (we only collected one trace for each scenario).

$$NMSE = \frac{\sum_{i=1}^{N}\left(X_i - \hat{X}_i\right)^2}{\sum_{i=1}^{N} X_i^2} \qquad (8)$$

As far as window size of known samples is concerned, we used always a fixed size of 30 samples for DOUjd and MOUjd. Regarding to Long and Sikdar model we tested three different window ranges, including both 3<N<10 and 10<N<100 ranges they used in [12].

With respect to smoothing of raw samples, we applied two different smoothing kernel functions: uniform and Triweight. Table 2 exhibits NMSE for forecasted samples using a uniform smoothed RSSI samples, which is the easiest smoothing kernel to calculate.

Table 2: Results of NMSE using Uniform smoothing Kernel

Scenario	< Uniform Kernel> Smoothing · 10^{-5}		
	DOUjd	**MOUjd**	**Long-Sikdar**
	N=30	N=30	3<N<10
1	*6,5235*	*8,7533*	*18,1618*
2	*3,1341*	*4,3299*	*9,4061*
3	*4,0717*	*9,5697*	*8,6834*
4	*5,7306*	*9,7304*	*13,917*
5	*32,158*	*91,595*	*65,877*
6	*0,0045*	*0,0071*	*0,0156*
7	*12,254*	*30,437*	*29,556*
8	*9,8362*	*15,824*	*21,7*

Table 3 exhibits NMSE for forecasted samples using a Triweight smoothing to data, which according to [12], gives the best performance for Long and Sikdar model.

Table 3: Results of NMSE using Triweight smoothing Kernel

Scenario	<Triweight Kernel> Smoothing · 10^{-5}		
	DOUjd	**MOUjd**	**Long-Sikdar**
	N=30	N=30	3<N<10
1	*13,981*	*17,035*	*42,606*
2	*6,2907*	*7,5022*	*21,686*
3	*5,9905*	*10,427*	*22,56*
4	*6,6289*	*17,115*	*32,129*
5	*97,93*	*186,19*	*90,811*
6	*0,0243*	*0,2701*	*0,03754*
7	*23,886*	*30,337*	*58,567*
8	*14,616*	*18,714*	*67,396*

In both tables the minimum NMSE value of each row is indicated by a gray shading. It is shown that with uniform kernel smaller NMSE was obtained by DOUjd.

When we used triweight kernel (Table 3) DOUjd also got the best results, except for scenario 5, where Long and Sikdar model was the best model. In the other hand, NMSE published in [12] was always very near to 2·10^{-6}. It can be noted that Table 2 and Table 3 show a wider range of NMSE (between 5·10^{-3} and 5·10^{-8}) for both methods. This is because of our selected scenarioes were heterogeneous and realistics considering tipycal situations of real life. On the contrary in [12] only a simple scenario was selected. No error aproximation was published in [7], so we have no reference of MOUjd behaviour. In our scenarioes it had an irregular performance; having an intermediate efficiency in all scenarioes except for 3rd, 5th and 7th with uniform smoothing and 5th and 6th with triweight smoothing, where had the worst behaviour. Scenario 6 (eBook) supplies the lowest NMSE. For Long and Sikdar model we studied an overlapped version 3<N<100, though with no significative result. But in all cases the best behaviour was reached with 3<N<10. This is the reason we only show this values in Tables 2 and 3.

In Fig. 5 to 12 we present graphically the evolution of RSSI with the different samples in order to enrich the conprehension of results of Tables 2 and 3. In each figure we present two graphics: one for the uniform smoothing funtion kernel and another for the Triweigth smoothing function kernel. In each graphic four rows are depicted: one for the smooth signal, and three more for each model (Long and Sikdar, MOUjd and DOUjd). Let us note that the value of NMSE represented is the original while in Tables 2 and 3 we scaled it in order to better understand its values.

In Fig. 5 is shown that the best results is obtained with our method (DOUjd) independently of the smooth kernel function used. The Long and Sikdar does not adapt well to sudden variation (for example around 8000).

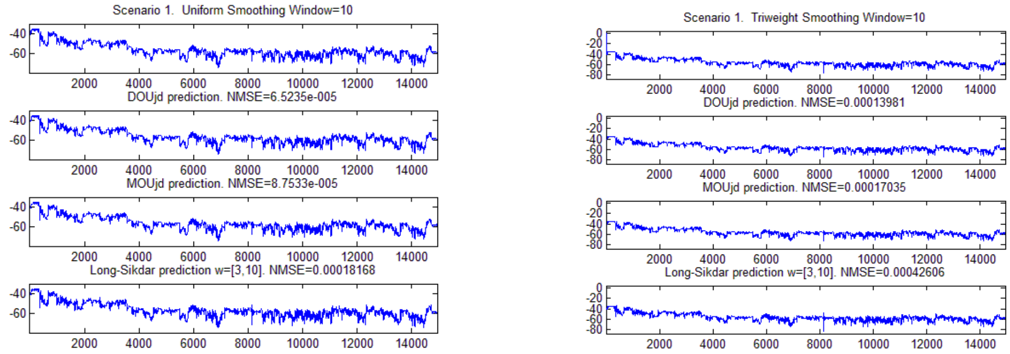

Figure 5. Graphics results of Scenario 1.

When the WiFi terminal is static the results of Fig. 6 are similar to Fig. 5, but with less differences between DOUjd and MOUjd, but for Long and Sikdar model the differences are mantained with Triweigth kernel. For uniform kernel the diferences are reduced among the three models due probably to there are less sudden variation of RSSI respecting to the scenario 1 where the WiFi Terminal was moving.

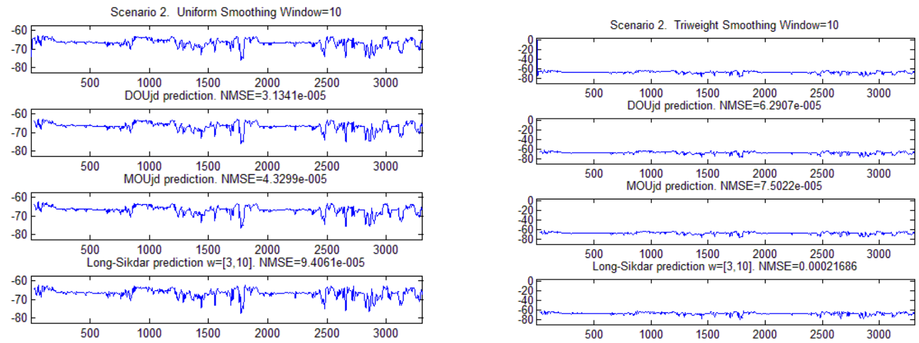

Figure 6. Graphics results of Scenario 2.

In scenario 3 the Long and Sikdar model is better than MOUjd for uniform kernel but it is double worse than MOUjd for Triweigth kernel. MOUjd is double worse than DOUjd.

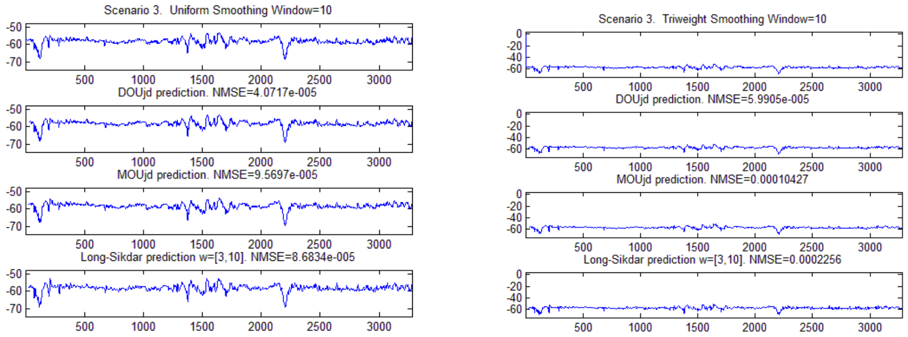

Figure 7. Graphics results of Scenario 3.

The results in scenario 4 (also the WiFi terminal is static) is very similar than the results for scenario 3 in which the WiFi terminal is static. This means that transmission power has little influence on the results.

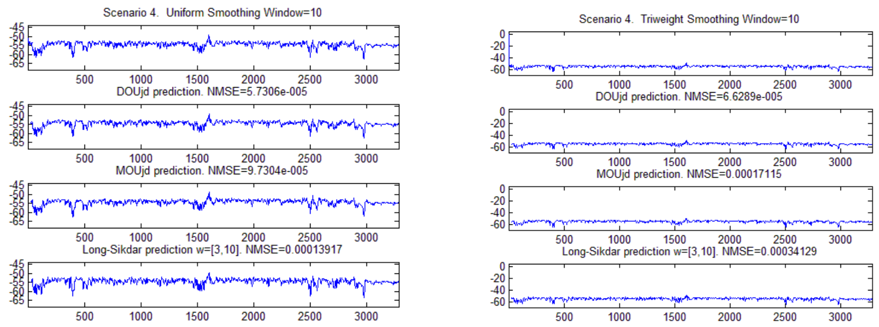

Figure 8. Graphics results of Scenario 4.

In scenario 5 the WiFi terminal is approaching to the AP. In this scenario, the MOUjd is double worse than DOUjd and than Long and Sikdar model, and this last one is double worse than DOUjd. But a curious effect is that with Triweigth kernel, the Long and Sikdar is a little better than DOUjd, but MOUjd is again double worse than Long and Sikdar model.

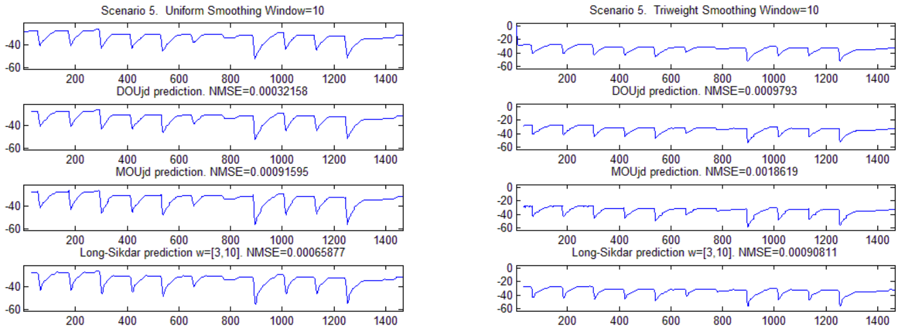

Figure 9. Graphics results of Scenario 5.

In scenario 6 the WiFi terminal (eBook) is static. RSSI captured by eBook exhibits an extremely reduced variability. This seems to be coherent if we consider eBook WiFi interface is designed for low speed text transmission, so sudden RSSI changes are not relevant. Moreover, eBook priority is battery saving. In this scenario DOUjd is much better than the other models for both kernels.

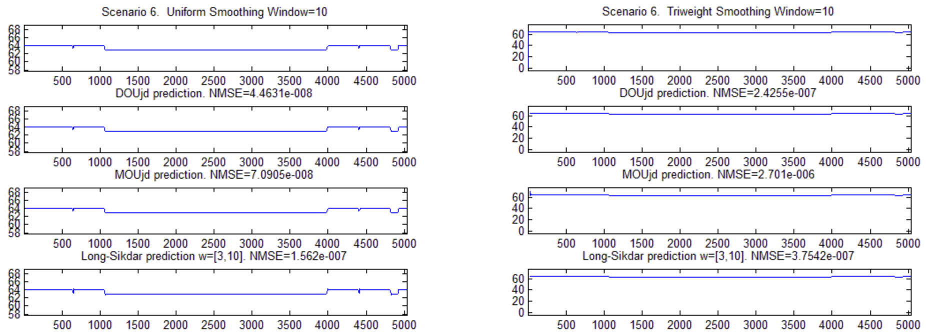

Figure 10. Graphics results of Scenario 6.

In scenario 7 we changed the eBook for a traditional Laptop using Linux operating system and its default monitoring software. In this case the best method is DOUjd.

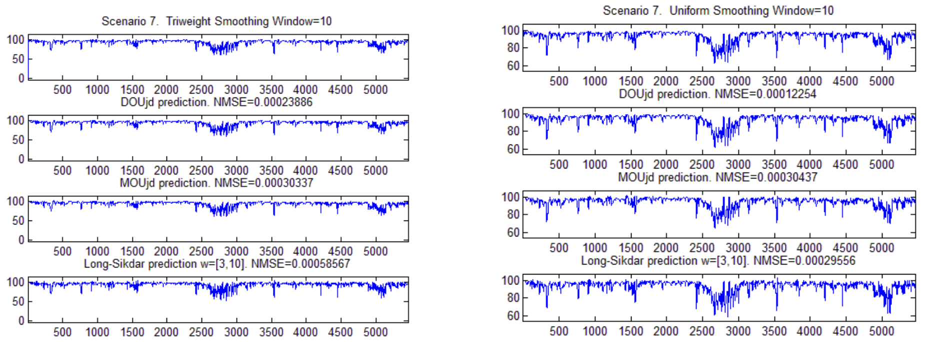

Figure 11. Graphics results of Scenario 7.

The difference between scenario 7 and 8 is that we changed the operating system and the monitoring Software (in scenario 8 it is used NetStumbler). The results are qualitatively the same as in the scenario 7.

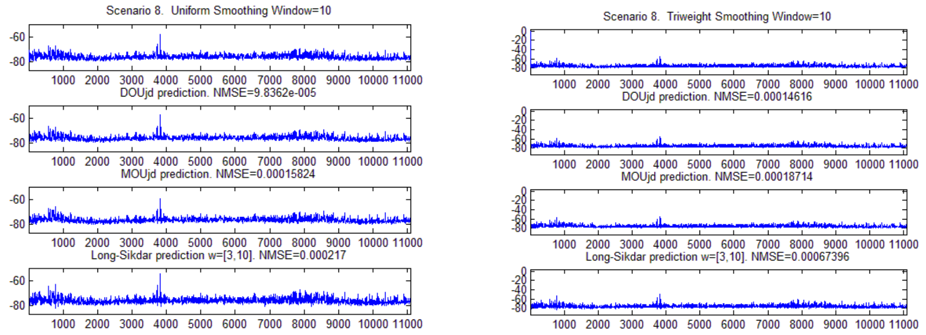

Figure 12. Graphics results of Scenario 8.

In summary, DOUjd was more efficient in a wider range of environments than the other methods. Despite the fact its computational complexity is equal than MOUjd method (about O(n^2) operations) and minor than Long and Sikdar method (about O(n^3) operations), our method does not compute square roots and it does not generate randoms numbers as the other two methods do, reducing the time to compute results. Though MOUjd is a similar method, had the worst behaviour in general. And it is to be expected it has everytime an irregular performance due to its dependence on stochastic terms. Triweight kernel for raw RSSI smoothing, which was suggested in [12] for improving the performance of Long and Sikdar model, did not seem to reduce NMSE of any model in our scenarioes, not even using Long and Sikdar model (it was the best method only in scenario 5, just with this kernel).

5. Conclusion

In this paper we have presented a prediction model of RSSI values (named *DOUjd*) considering heterogeneous devices, drivers, operating systems and monitoring software to take into account realistic and heterogeneous scenarios. We compared it with other recent models (MOUjd and Long and Sikdar models). All the models try to find the tendency of the RSSI curve. All of them smooth previously the raw signal. Precisely, the scenario 6 (eBook) shows that in the case the driver makes an early smooth of the signal the NMSE decreases. This could seem an advantage, but it is not a reflex of the good work of the models, because if we choose a driver that does not make this early smooth the models will work differently (as we have shown). All the models try to eliminate the sudden jump RSSI values.

As much DOUjd, as MOUjd and Long and Sikdar models try to predict succesive RSSI data values adjusting them through regression analysis to a window of well-known past RSSI samples. Long and Sikdar's model handles a dynamic window whose size keeps increasing while forecast error in the following predicted sample does not exceed some threshold. Thus they achieve in forecasting the general trend of signal, though allowing some samples leak out that tendency. Drawback in this window dynamism is that first predicted samples, after a size reset, will be characterized by a higher forecast error. On the other hand, DOUjd manages a fixed size window for predicting the general RSSI trend. But it is also flexible enough by allowing several samples to deviate from the general trend. Our method peculiarity is that trend deviated samples (jumps) are included in the well-known RSSI data window, which will generate the regression coefficients; meanwhile, in Long and Sikdar model, forecasted values are the samples wich can deviate from the general signal trend. Althougth the main objective of MOUjd is to eliminate the effect of the sudden jump RSSI values on the tendency of the RSSI curve, in the experimental results in our scenarios this is not clearly appreciated.

With the learnt lessons we are preparing a new version of our prediction model to apply it to efficient handover management.

References

[1] Goussevskaia O.,Halldórsson M.M., Wattenhofer R. “Algorithms for Wireless Capacity”. IEEE/ACM Transactions on Networking, Vol. 22, Nº. 3. pp. 745-755. 2014. DOI: 10.1109/TNET.2013.2258036

[2] Oliveira T., Raju M., Agrawal D.P., “Accurate Distance Estimation Using Fuzzy based Combined RSSI/LQI Values in an Indoor Scenario: Experimental Verification”. Network Protocols and Algorithms. Vol. 4, Issue 4. pp. 174-199. 2012. DOI: 10.5296/npa.v4i4.2173

[3] Elbes M., Al-Fuqaha A., Anan M., “A Precise Indoor Localization Approach based on Particle Filter and Dynamic Exclusion Techniques”. Network Protocols and Algorithms. Vol. 5, Issue 2. pp. 50-71. 2013. DOI: 10.5296/npa.v5i2.3717

[4] Lin S., Zhang J., Zhou G., Gu L., He T., Stankovic J.A., “ATPC: adaptive transmission power control for wireless sensor networks”, The Fourth International Conference on Embedded Networked Sensor Systems (SenSys’06). Boulder, Colorado, USA, November 1–3, 2006. DOI: 10.1145/1182807.1182830

[5] Rondinone M., Ansari J., Riihijärvi J., Mähönen P., “Designing a reliable and stable link quality metric for wireless sensor networks, The workshop on Real-world Wireless Sensor Networks (REALWSN '08). Glasgow, United Kingdom, April 1, 2008. pp. 6-10. DOI: 10.1145/1435473.1435476

[6] Suárez A., Elbatsh K. A., Macías E., "Gradient RSSI Filter and Predictor for Wireless Network Algorithms and Protocols". Network Protocols and Algorithms. Vol. 2, Issue 2. pp. 1-26. 2010. DOI: 10.5296/npa.v2i2.414

[7] Chin E., Chieng D., The V., Natkaniec M., Loziak K., Gozdecki J., "Wireless Link Prediction and Triggering using Modified Ornstein-Uhlenbeck Jump Diffusion Process". Wireless Networks. Vol. 20, Issue 3. pp. 379-396. 2014. DOI: 10.1007/s11276-013-0610-0

[8] Smith, W., "On the Simulation and Estimation of the Mean-Reverting Ornstein-Uhlenbeck Process. Especially as Applied to Commodities Markets and Modelling". Available at commoditymodels.com

[9] Sitmo. Calibrating the Ornstein-Uhlenbeck (Vasicek) Model. Available at: http://www.sitmo.com/article/calibrating-the-ornstein-uhlenbeck-model/ (last accessed September 30, 2014)

[10] Cont, R., Tankov, P., "Financial modelling with jump processes", in *Chapman & Hall/CRC Financial Mathematics Series*. Chapman & Hall/CRC, Boca Raton, FL, 2004.

[11] Hanson, F.B., "Applied Stochastic Processes and Control for Jump-Diffusions: Modeling, Analysis, and Computation, Advances in Design and Control", Ed. Society for Industrial and Applied Mathematics, 2007.

[12] Long X., Sikdar B., "A real-time algorithm for long range signal strength prediction in wireless networks", IEEE Wireless Communications and Networking Conference (WCNC 2008). Las Vegas, Nevado, USA, March 31-April 3, 2008, pp. 1120-1125. DOI: 10.1109/WCNC.2008.202

[13] Sadiq A. S., Bakar K. A., Ghafoor K. Z., Lloret J. ,Mirjalili S., "A Smart Handover Prediction System based on Curve Fitting Model for Fast Mobile IPv6 in Wireless Networks". International Journal of Communication Systems. Vol. 27. pp. 969–990. 2014. DOI: 10.1002/dac.2386

[14] IEEE Std. 802.11 WG, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.

[15] Bardwell, J., "You Believe You Understand What You Think I Said…The Truth About 802.11 Signal And Noise Metrics". Available at http://n-cg.net/ncgpdf/WiFi_SignalValues.pdf

[16] iwconfig, iwlist, Wireless-tools. Available at: http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html (last accessed September 30, 2014)

[17] IEEE Standard for Information technology- Std. 802.11n Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Amendment 5: Enhancements for Higher Throughput. 2009.

[18] IEEE Std. 802.11 WG Std 802.11a, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High speed Physical Layer in the 5 GHz Band, 1999.

[19] IEEE Std. 802.11 WG Std. 802.11b, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Further Higher-Speed Physical Layer Extension in the 2.4 GHz Band 1999.

[20] IEEE Std. 802.11 WG Std. 802.11g, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Further Higher-Speed Physical Layer Extension in the 2.4 GHz Band, 2003.

[21] Bri D., Fernández-Diego, M., García M., Ramos F., Lloret J. "How the Weather Impacts on the Performance of an Outdoor WLAN". IEEE Communications Letters. Vol. 16, Issue 8. pp. 1184-1187. 2012. DOI: 10.1109/LCOMM.2012.060812.120090

[22] RadioTap. Available at: http://www.radiotap.org/ (last accessed September 30, 2014)

Copyright Disclaimer

Copyright reserved by the author(s).

This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/3.0/).