

Wireless Netw (2014) 20:379–396
DOI 10.1007/s11276-013-0610-0

Wireless link prediction and triggering using modified Ornstein–Uhlenbeck jump diffusion process

Eric Chin · David Chieng · Victor Teh ·
Marek Natkaniec · Krzysztof Loziak ·
Janusz Gozdecki

Published online: 2 July 2013

© The Author(s) 2013. This article is published with open access at Springerlink.com

Abstract Through time domain observation, typical wireless signal strength values seems to exhibit some forms of mean-reverting and discontinuous “jumps” behaviour. Motivated by this fact, we propose a wireless link prediction and triggering (LPT) technique using a modified mean-reverting Ornstein–Uhlenbeck (OU) jump diffusion process. The proposed technique which we refer as OU-LPT is an integral component of wireless mesh network monitoring system developed by ICT FP7 CARrier grade wireless MESH Network project. In particular, we demonstrate how this technique can be applied in the context of wireless mesh networks to support link switching or handover in the event of predicted link degradation or failure. The proposed technique has also been implemented and evaluated in a real-time experimental testbed. The results show that OU-LPT technique can significantly enhance the reliability of wireless links by reducing the rate of false triggers compared to a conventional linear

prediction technique and therefore offers a new direction on how wireless link prediction, triggering and switching process can be conducted in the future.

Keywords Wireless mesh networks · Monitoring system · Link prediction · Link triggering · Data analysis

1 Introduction

Monitoring of a wireless link is a tough challenge due to the nature of the wireless link which is constantly affected by interference and temporary fading. An efficient and reliable network monitoring system is generally expected to collect information regarding current system configuration, observe current values of parameters influencing performance metrics, detect abnormal behaviour of a node or link and in some cases, predict the performance degradation events. Accurate and timely prediction is critical to ensure that there is sufficient time for mitigation actions such as self (re)configuration or healing to take place [3]. In wireless multi-hop [4] or mesh [5] environment, the behaviour of links particularly those which are closer to a gateway is becoming the primary concern as they carry the traffic of other nodes further down the hops. Through time domain observation typical wireless signals’ signal strength seem to exhibit some forms of mean-reverting behaviour (converging towards a long term mean) as well as discontinuous “jumps” (missing values for a certain period of time). Therefore it is natural for us to look at models with these properties. One such stochastic model which we are considering in this paper is the Ornstein–Uhlenbeck (OU) diffusion process which was first applied in physics [7] to describe Brownian motion of particles suspended in a fluid with friction. In this paper due to the inherent jump

E. Chin · D. Chieng · V. Teh
BT Innovate and Design, Adastral Park, Martlesham Heath,
Ipswich IP5 3RE, UK
e-mail: eric.chin@bt.com

D. Chieng
e-mail: david.chieng@bt.com

V. Teh
e-mail: victorteh@gmail.com

M. Natkaniec (✉) · K. Loziak · J. Gozdecki
AGH University of Science and Technology,
al. Mickiewicza 30, 30-059 Kraków, Poland
e-mail: natkanie@kt.agh.edu.pl

K. Loziak
e-mail: kloziak@kt.agh.edu.pl

J. Gozdecki
e-mail: gozdecki@kt.agh.edu.pl

properties of wireless signal strength or received signal strength indicator (RSSI) values, we propose the modelling of this behaviour using a modified Ornstein–Uhlenbeck jump diffusion process. The proposed technique is an integral part of the monitoring system developed by ICT FP7 CARMEN (CARrier grade wireless MESH Network) project [1, 2, 6, 8].

The remainder of this paper is organised as follows. Section 2 discusses the related works and motivations behind this research. Section 3 presents the modelling, and calibration of the proposed method and the novel link prediction and triggering algorithms. Section 4 shows the OU-LPT analysis. It also demonstrates the design and real-time implementation of OU-LPT technique. Finally conclusions are drawn in Sect. 5.

2 Related work

Many studies have been done on wireless link monitoring in general and each of them provided us with different approaches, methods or techniques. On work to improve monitoring accuracy, the efficient and accurate link-quality monitor (EAR) developed by [9] exploits three complementary measurement schemes namely: passive, cooperative and active monitoring. It maximizes the measurement accuracy by dynamically and adaptively adopting one of the above mentioned measurement schemes. For link quality monitoring, many are using signal to noise ratio (SNR) or RSSI measurement as quality measure [10–13]. According to MadWiFi driver [14], the reported RSSI for each frame is actually equivalent to the SNR and therefore the terms can actually be used interchangeably except that the definition of RSSI usually varies between vendors. The work in [10] confirms that the SNR is a very good indicator for choosing the optimum bit rate for IEEE802.11 [15] in general when trained on a particular link. Authors in [13] found that RSSI is an appropriate metric for quantifying the link quality and channel dynamics when compared with the value measured by a spectrum analyzer. The work in [12] proposes an accurate, low-complexity, on-line prediction mechanism for the long range prediction of wireless link quality. Similarly this work also uses RSSI as the basic measure for signal strength. Here the past measurements of the received signal strength are employed and then through segmentation, filtering and regression process, the future trend in the received signal strength is forecasted. [11] proposes XCoPred, which is a pattern matching based scheme to predict link quality variations. The nodes monitor and store the links SNR values to their neighbours in order to obtain time series of SNR measurements. When a prediction on the future state of a link is required, the node looks for similar SNR patterns to the current situation in

the past using a cross correlation function. Mesh-Mon [13] on the other hand aims to actively cooperate and predict, detect, diagnose and resolve network problems in a scalable manner. It is independent of the underlying routing protocol and can operate even if the mesh routing protocol fails completely. In our work, we propose a novel technique that takes advantage of mean-reverting behaviour of a RSSI as well as its discontinuous “jumps” characteristic.

As revealed in [10], it is understood that though RSSI or SNR alone is good enough for a single link, it may not achieve sufficient accuracy in deciding the end-to-end or network wide transmission quality. In such situation, other cross layer metrics such as (MAC/IP layer) latency, throughput and loss may provide a more accurate means to evaluate the quality of a link. Metrics such as expected transmission count (ETX) and expected transmission time (ETT) have been widely proposed to support routing decision in wireless mesh [16]. However these metrics depend very much on the types of application and hence pose additional complexities when performing prediction. First and foremost, the monitoring system would need to know exact traffic pattern of the sender, then there is a foreseen challenge on trigger timeliness since a specific period is required to collect, compute and analyze cross-layer frame information. In this paper therefore, we only focus on the SNR/RSSI as it generally provides a reasonably good indication on the quality of the link without having to know the traffic characteristics, patterns or distribution. Despite saying that, the proposed link prediction and triggering technique can be applied on any desired metric such as throughput, delay, jitter or loss rate as long as it exhibits some forms of mean-reverting behaviour and discontinuous “jump”.

3 Link prediction and triggering with OU diffusion process (OU-LPT)

To make a reliable forecast of local and neighbouring mesh links, we propose a diffusion process models for a selected window size of a series of RSSI values. The prediction can be applied to any channel info received from the neighbouring radios. Instead of using statistical time series modelling which involves comprehensive model identification process and then parameters estimation that are numerically intensive, we propose a much more simpler and effective way to estimate diffusion process model parameters from historical data.

3.1 Ornstein–Uhlenbeck jump diffusion process

Figure 1 shows both the time series of raw and smoothed RSSI values of a typical WLAN link with a time step of

100 ms. The latter is smoothen via moving average method with an average of 10 values in order to reduce short term fluctuations. By observing the time domain behaviour, the RSSI data seems to exhibit some forms of mean-reverting behaviour as well as discontinuous “jumps”, therefore it is natural for us to look at one such stochastic model such as the Ornstein–Uhlenbeck (OU) diffusion process.

We propose the modelling of such behaviour using a modified mean-reverting diffusion process called the OU jump diffusion process, $OU(\kappa, \theta, \sigma, J)$, defined by the stochastic differential equation (SDE):

$$dX_t = \kappa(\theta - X_t)dt + \sigma dW_t + \log J_t dN_t \tag{1}$$

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.

$$X_{t+\ell\Delta t} = X_t e^{-\kappa\ell\Delta t} + \left(\theta + \frac{\lambda\mu_J}{\kappa}\right)(1 - e^{-\kappa\ell\Delta t}) + \sigma\sqrt{\frac{1 - e^{-2\kappa\ell\Delta t}}{2\kappa}}Z_1 + \sqrt{\frac{\lambda}{2\kappa}(\mu_J^2 + \sigma_J^2)}Z_2 \tag{2}$$

and given X_t it has expectation $E(X_{t+\ell\Delta t}) = X_t e^{-\kappa\ell\Delta t} + \left(\theta + \frac{\lambda\mu_J}{\kappa}\right)(1 - e^{-\kappa\ell\Delta t})$, and variance $Var(X_{t+\ell\Delta t}) = \sigma^2\left(\frac{1 - e^{-2\kappa\ell\Delta t}}{2\kappa}\right) + \frac{\lambda}{2\kappa}(\mu_J^2 + \sigma_J^2)$ where $Z_1, Z_2 \sim N(0,1)$ and Z_1, Z_2 are independent.

3.2 Model calibration

In order to calibrate the parameters $\kappa, \theta, \sigma, \lambda, \mu_J$ and σ_J , we can subdivide the OU jump diffusion process as follows:

$$dX_t = \begin{cases} \kappa(\theta - X_t)dt + \sigma dW + \log J_t dN_t & \text{if Poisson event occurs} \\ \kappa(\theta - X_t)dt + \sigma dW_t & \text{if Poisson event does not occurs} \end{cases}$$

The process dN_t is a Poisson process with parameter λ such that

$$dN_t = \begin{cases} 1 & \text{with probability } \lambda dt \\ 0 & \text{with probability } 1 - \lambda dt \end{cases}$$

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. Using the analysis given by [18] for each forecast step ahead $\ell \geq 1$ and a constant time step Δt we can solve the SDE as

To begin with, the jump diffusion model in its simplest form needs an estimate of probability of jump, measured by λ , and its size J_t . However, this can be made more complicated by having a distribution for J_t . Given that we have an array of parameters to estimate and if we were to set up a maximum likelihood method for the full model (mixing jumps and diffusion), it may be hard for the algorithm to distinguish what are jumps, and what are

Fig. 1 Time series of raw and smoothed WLAN RSSI values

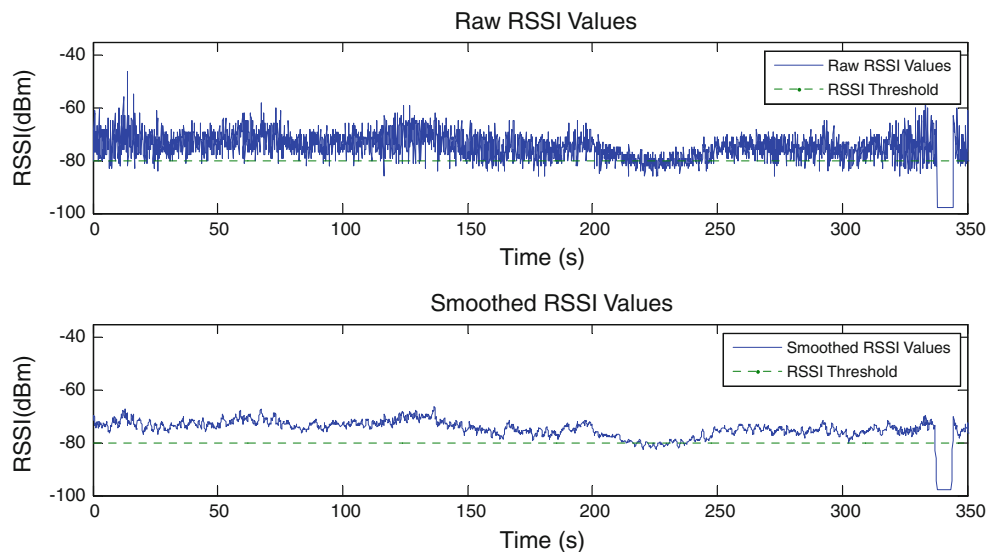
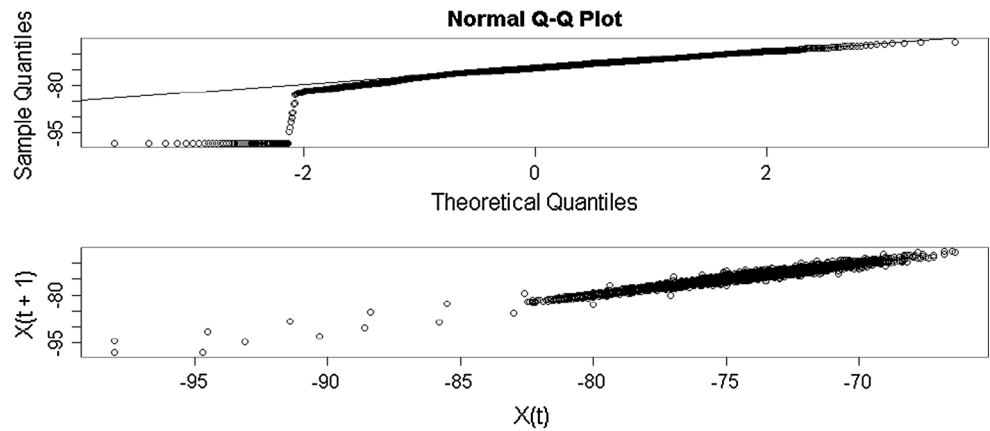


Fig. 2 Q-Q normal probability plot of returns (*top*) and scatter plot of $X_{t+\Delta t}$ against X_t (*bottom*)



diffusions. Hence there is a need for us to subdivide the parameter estimation of jump components and mean reversion diffusion process into two parts.

As seen from the normal probability plot of returns $r_t = X_{t+\Delta t} - X_t$ where $\Delta t = 0.1$ s in Fig. 2, the existence of fat tails suggest the probability of rare events occurring is higher than predicted by a Gaussian distribution. In addition from the scatter plot of $X_{t+\Delta t}$ against X_t , a linear fit to describe their interaction is inappropriate with the presence of jumps. In addition the histogram of the returns together with a fitted normal density (see Fig. 3) shows there is a significant existence of fat tails in which the returns data are not normally distributed.

In order to extract the jump components from a series of returns $r_t = X_{t+\Delta t} - X_t$, we can use the following pseudo-code:

Begin

Set $R = \{r_1, r_2, \dots, r_N\}$ and its complement $R^C = \phi$ where N is the number of observations.

Repeat

- Find the mean \bar{r} and standard deviation σ_r of the set R
- For all elements in the set R , filter out the return r_i if $|r_i - \bar{r}| > 3\sigma_r$. Set the filtered out set $R^C = R^C \cup \{r_i\}$ and $R = R - \{r_i\}$

Until no further returns are filtered.

End

From the output of the filtered set R^C , we can estimate the frequency of jumps $\hat{\lambda}$, mean $\hat{\mu}_J$ and variance $\hat{\sigma}_J^2$ as

$$\hat{\lambda} = \left(\frac{|R^C|}{|R| + |R^C|} \right) \Delta t, \hat{\mu}_J = \bar{r}^C \text{ and } \hat{\sigma}_J^2 = \sigma_{r^C}^2 \tag{3}$$

where $|R^C|$ and $|R|$ are the cardinal numbers of the filtered out series R^C and the filtered series R respectively, \bar{r}^C and $\sigma_{r^C}^2$ are the mean and variance of the set R^C respectively.

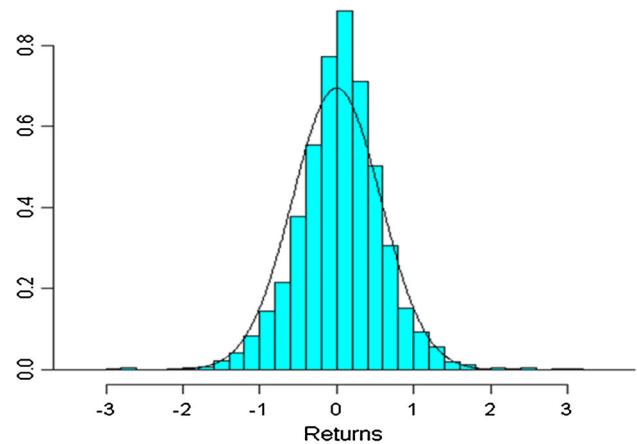


Fig. 3 Histogram of returns $r_t = X_{t+\Delta t} - X_t$ with a fitted normal density is shown overlaid

Once we have extracted out the jump components from the original series, we can see from the normal probability plot of filtered returns in Fig. 5 that there is a high proportion of residuals being on the straight line passing through zero. The histogram also shows that most of data that constitute fat tails have been removed. This shows that the normal plot of filtered residuals is well behaved.

In addition from the scatter plot of filtered $X_{t+\Delta t}$ against X_t in Fig. 4, we can deduce that there is a strong linear relationship between them, and hence we can fit a linear model to describe their interaction. In order to estimate the remaining mean reversion parameters we can use a subset of the filtered series $\{X_t\}$ to estimate the parameters for each time step. Having identified which part of the original series have jump components or statistically significant jumps, we can then extract out a subset of the original series where the returns are continuous which follow an OU process given as:

$$dX_t = \kappa(\theta - X_t)dt + \sigma dW_t \tag{4}$$

Taking note that

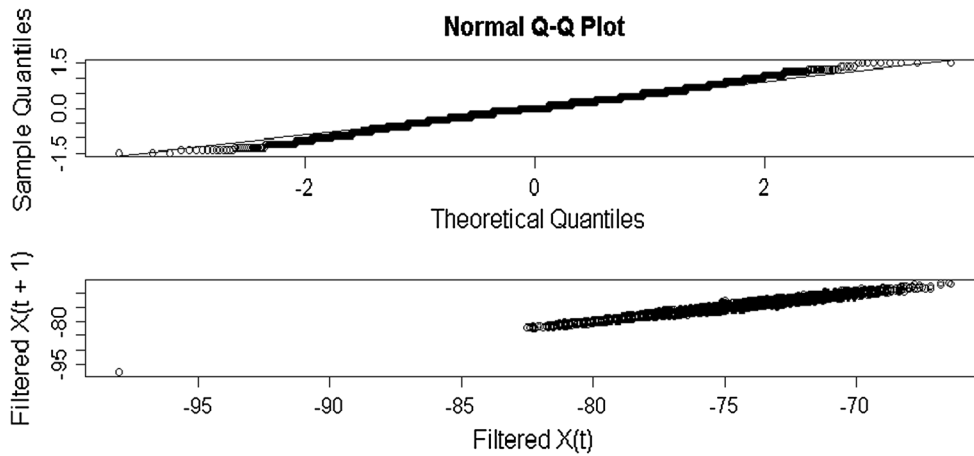


Fig. 4 Normal probability plot of filtered returns (top) and scatter plot of filtered X_{t+1} against X_t (bottom)

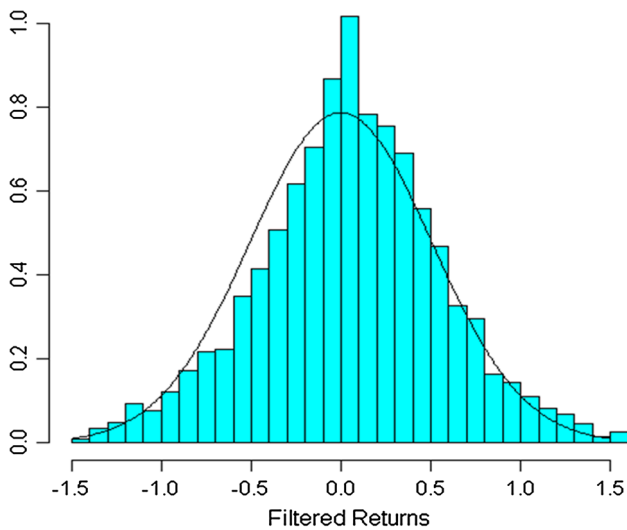


Fig. 5 Histogram of filtered returns with a fitted normal density is shown overlaid

$$d(e^{\kappa t} X_t) = \kappa e^{\kappa t} X_t dt + e^{\kappa t} dX_t + \frac{1}{2} \kappa^2 e^{\kappa t} X_t (dt)^2 + \dots \quad (5)$$

and from Ito’s lemma [19] such that

$$(dX_t)^2 = \sigma^2 dt, (dt)^v = o(1), v > 1 \quad (6)$$

for any arbitrary time step $\Delta t > 0$, the Ornstein–Uhlenbeck process has a unique solution

$$X_{t+\Delta t} = X_t e^{-\kappa \Delta t} + \theta(1 - e^{-\kappa \Delta t}) + \sigma \sqrt{\frac{1 - e^{-2\kappa \Delta t}}{2\kappa}} Z \quad (7)$$

where $Z \sim N(0,1)$ follows a standard normal distribution.

In this study, rather than accurately finding the parameter values using expensive maximum likelihood estimation method we can instead rely on simple regression analysis. As shown in Fig. 4, we can see that there is a strong linear relationship between X_{t+1} and X_t (we take

$\Delta t = 0.1$ s) for all t values. Hence the first step in our parameter estimation using regression analysis is to find the best fit of the RSSI time series $\{X_t\}$ to its past values in order to make future forecasts.

To begin with by taking $N > 2$ to be the size of the window for the series of data and Δt be the step size, we let the relationship between consecutive RSSI values $X_t, X_{t+\Delta t}, X_{t+2\Delta t}, \dots, X_T$

$$X_{t+\Delta t} = aX_t + b + \varepsilon_t, \varepsilon_t \sim N(0, \sigma_\varepsilon^2) \quad (8)$$

where $T = t + N\Delta t$, a and b are the regression parameters, ε_t is normally distributed and is independent and identically distributed.

By comparing the relationship between the linear fit and the OU process model, the parameters can be equated as

$$a = e^{-\kappa \Delta t}, b = \theta(1 - e^{-\kappa \Delta t}), \sigma_\varepsilon = \sigma \sqrt{\frac{1 - e^{-2\kappa \Delta t}}{2\kappa}} \quad (9)$$

Given we require $\kappa > 0$ and provided $a \in (0,1)$ we can then set the jump-diffusion process model with the formulas

$$\theta = \frac{b}{1 - a}, \kappa = -\frac{\log a}{\Delta t}, \sigma = \sigma_\varepsilon \sqrt{\frac{-2 \log a}{(1 - a^2)\Delta t}} \quad (10)$$

In order to find the optimal values \hat{a} and \hat{b} we can solve the following least-squares regression optimization problem subject to a bound constraint:

$$P \begin{cases} \text{minimize} & \sum_{i=t+\Delta t}^T (X_i - aX_{i-\Delta t} - b)^2 \\ & a, b \\ \text{subject to} & 0 < a < 1. \end{cases}$$

Instead of using computationally expensive optimization subroutines such as L-BFGS-B method [17] to solve problem P iteratively, we can utilise least-squares regression analysis by first calculating the following quantities

$$\begin{aligned}
 S_x &= \sum_{i=t+\Delta t}^T X_{i-\Delta t}, S_y = \sum_{i=t+\Delta t}^T X_i, S_{xx} = \sum_{i=t+\Delta t}^T X_{i-\Delta t}^2, S_{xy} \\
 &= \sum_{i=t+\Delta t}^T X_{i-\Delta t}X_i, S_{yy} = \sum_{i=t+\Delta t}^T X_i^2
 \end{aligned}
 \tag{11}$$

and set

$$\hat{a}_0 = \frac{NS_{xy} - S_x S_y}{NS_{xx} - S_x^2}
 \tag{12}$$

Hence the optimal values \hat{a} and \hat{b} can be estimated as follows:

$$\hat{a} = \begin{cases} \hat{a}_0 & \text{if } 0 < \hat{a}_0 < 1 \\ \varepsilon & \text{if } \hat{a}_0 \leq 0 \\ 1 - \varepsilon & \text{otherwise} \end{cases} \quad \text{and } \hat{b} = \frac{S_y - \hat{a}S_x}{N}
 \tag{13}$$

where $\varepsilon \in (0, 1)$.

In addition under the assumption that the error term has a constant variance, once we have found the optimal values \hat{a} and \hat{b} , the estimated standard deviation of the error term is

$$\hat{\sigma}_\varepsilon = \sqrt{\frac{NS_{yy} - S_y^2 - \hat{a}(NS_{xy} - S_x S_y)}{N(N - 2)}}
 \tag{14}$$

Note that if $\kappa\Delta t \ll 1$ then we can approximate

$$a \approx 1 - \kappa\Delta t, b \approx \theta\kappa\Delta t, \sigma_\varepsilon \approx \sigma\sqrt{\Delta t}
 \tag{15}$$

with errors of order $O((\kappa\Delta t)^2)$. Hence the mean reversion model parameters can be approximated as

$$\hat{\theta} = \frac{\hat{b}}{1 - \hat{a}}, \hat{\kappa} = \frac{1 - \hat{a}}{\Delta t}, \hat{\sigma} = \frac{\hat{\sigma}_\varepsilon}{\sqrt{\Delta t}}
 \tag{16}$$

Collectively we can then write

$$\begin{aligned}
 \hat{X}_{t+\Delta t} &= X_t e^{-\hat{\kappa}\Delta t} + \left(\hat{\theta} + \frac{\hat{\lambda}\hat{\mu}_I}{\hat{\kappa}} \right) (1 - e^{-\hat{\kappa}\Delta t}) \\
 &\quad + \hat{\sigma} \sqrt{\frac{1 - e^{-2\hat{\kappa}\Delta t}}{2\hat{\kappa}}} Z_1 + \sqrt{\frac{\hat{\lambda}}{2\hat{\kappa}}} (\hat{\mu}_J^2 + \hat{\sigma}_J^2) Z_2
 \end{aligned}
 \tag{17}$$

where $Z_1, Z_2 \sim N(0,1)$ and Z_1, Z_2 are independent.

3.3 Prediction algorithm

Once the parameter values $\hat{\kappa}, \hat{\theta}, \hat{\sigma}, \hat{\lambda}, \hat{\mu}_J$ and $\hat{\sigma}_J$ are estimated, we can then deduce the estimated forecast $\hat{X}_{t+\ell\Delta t}$ follows

$$\frac{\hat{X}_{t+\ell\Delta t} - E(\hat{X}_{t+\ell\Delta t})}{\sqrt{\text{Var}(\hat{X}_{t+\ell\Delta t})}} \sim N(0, 1)
 \tag{18}$$

Assuming the current mesh node has the knowledge on all (or some of) the neighbouring mesh nodes' channel's

RSSI of which it can form a link with. The mesh node would only issue a trigger when its present forecasted RSSI value falls below its threshold value, and the forecasted RSSI value of a target neighbouring mesh node exceeds its threshold value. By denoting the neighbouring mesh nodes RSSI values as $Y_t^{(i)}$ where $i=1, 2, \dots, M$, where M is the total number of neighbouring mesh nodes (or mesh radios in multi-radio case), the current mesh node would only issue a trigger when it current link's RSSI

$$\hat{X}_{t+\ell\Delta t} \leq \bar{X} \text{ and } \hat{Y}_{t+\ell\Delta t}^{(j)} > \bar{Y}^{(j)}
 \tag{19}$$

where the index j is defined as

$$j = \{i : \max\{\hat{Y}_{t+\ell\Delta t}^{(i)} - \bar{Y}^{(i)}, 0\}, \quad i = 1, 2, \dots, M\}
 \tag{20}$$

where \bar{X} is the current link RSSI threshold representing the minimal QoS it must support in order to operate successfully, $\bar{Y}^{(i)}$ is the i th neighbouring RSSI threshold value and $\hat{Y}_{t+\ell\Delta t}^{(j)}$ is the predicted RSSI value of the j th neighbouring mesh node if which it could form a new link with. The criteria given in (19)–(20) denotes that the OU-LPT method would only choose the “best” neighbouring mesh node. On the other hand if there is no better mesh node, then the scheme will not trigger a link handover event.

The Link Going Down (LGD) event is introduced to help wireless nodes to prepare for link handover or switching prior to Link Down (LD) so that switching delays and service interruptions can be minimized. Based on the forecasted RSSI values of the current link and in order to minimize the error of decision making, wireless card manufacturers like Intel [20] would introduce a protection margin for LGD (or hysteresis factor) $\Delta_x^{GD} \geq 0$. The purpose of having this protection margin is to augment it to the RSSI threshold value, \bar{X} so that the current link has an enhanced threshold value, $\bar{X} + \Delta_x^{GD}$ to ensure a better QoS. If the forecasted RSSI value is greater than the enhanced threshold value, then the system would not trigger a link handover to another mesh node. In the following Table 1 we list the trigger thresholds that are being used in this report where $\Delta_x^U > \Delta_x^{CU} > \Delta_x^{GD} > 0$.

With this protection margin Δ_x^{GD} , and for a forecasted RSSI value $\bar{X}_{t+\ell\Delta t}$ the probability in making a trigger is defined as

Table 1 Thresholds for link handover trigger

Link-Up threshold (LU_TH)	$\bar{X} + \Delta_x^U$
Link-coming-up threshold (LCU_TH)	$\bar{X} + \Delta_x^{CU}$
Link-going-down threshold (LGD_TH)	$\bar{X} + \Delta_x^{GD}$
Link-down threshold (LD_TH)	\bar{X}

$$P(\hat{X}_{t+\ell\Delta t} \leq \bar{X} + \Delta_x^{GD}) = P\left(Z \leq \frac{\bar{X} + \Delta_x^{GD} - E(\hat{X}_{t+\ell\Delta t})}{\sqrt{Var(\hat{X}_{t+\ell\Delta t})}}\right) \tag{21}$$

and if

$$E(\hat{X}_{t+\ell\Delta t}) \leq \bar{X} + \Delta_x^{GD} \text{ and } P(\hat{X}_{t+\ell\Delta t} \leq \bar{X} + \Delta_x^{GD}) \geq \alpha \tag{22}$$

where $\alpha \in (0,1)$ is a margin error then the current mesh node will issue a trigger at time t to initiate a link handover to an alternative mesh node or radio.

In addition, for the forecasted RSSI values of neighbouring mesh nodes and for each of the i th node we also introduce a protection margin $\Delta_y^{(i)} \geq 0$ so as to minimize the error of false selection of a node for handover. By analogy with the probability of making a trigger for the mesh link, for each neighbouring mesh radios, we define the probability of selecting a new node:

$$P(\hat{Y}_{t+\ell\Delta t}^{(i)} \geq \bar{Y}^{(i)} + \Delta_{y^{(i)}}^{CU}) = P\left(Z \geq \frac{\bar{Y}^{(i)} + \Delta_{y^{(i)}}^{CU} - E(\hat{Y}_{t+\ell\Delta t}^{(i)})}{\sqrt{Var(\hat{Y}_{t+\ell\Delta t}^{(i)})}}\right), \tag{23}$$

$i = 1, 2, \dots, M$

and if

$$P(\hat{Y}_{t+\ell\Delta t}^{(i)} \geq \bar{Y}^{(i)} + \Delta_{y^{(i)}}^{CU}) \geq \beta \tag{24}$$

where $\beta \in (0,1)$ is a margin error, then the i th n can be selected to be the link handover target. By augmenting a protection margin to our OU-LPT method we can now redefine our criterion of a handover from a current mesh node to the j th mesh node at time t as:

$$\left\{ E(\hat{X}_{t+\ell\Delta t}) \leq \bar{X} + \Delta_x^{GD} \text{ and } P(\hat{X}_{t+\ell\Delta t} \leq \bar{X} + \Delta_x^{GD}) \geq \alpha \right\} \text{ and } \left\{ E(\hat{Y}_{t+\ell\Delta t}^{(j)}) \geq \bar{Y}^{(j)} + \Delta_{y^{(j)}}^{CU} \text{ and } P(\hat{Y}_{t+\ell\Delta t}^{(j)} \geq \bar{Y}^{(j)} + \Delta_{y^{(j)}}^{CU}) \geq \beta \right\} \tag{25}$$

where the index j is defined as

$$j = \left\{ i : \max \left\{ P(\hat{Y}_{t+\ell\Delta t}^{(i)} \geq \bar{Y}^{(i)} + \Delta_{y^{(i)}}^{CU}), \quad i = 1, 2, \dots, M \right\} \right\} \tag{26}$$

By analogy with statistical hypothesis testing, the procedure described above would lead us to commit a false trigger (or false positive) error. With this protection margin Δ_x , and for a forecasted RSSI value $\hat{X}_{t+\ell\Delta t}$, here we define the probability in making a false trigger (or false alarm) at time t as

$$P(\hat{X}_{t+\ell\Delta t} \leq \bar{X} + \Delta_x^{GD} | X_{t+\ell\Delta t} > \bar{X} + \Delta_x^{GD}) \tag{27}$$

where it is the error of committing a false trigger when the true RSSI value $X_{t+\ell\Delta t}$ is greater than the enhanced threshold requirement but the forecasted RSSI value, $\hat{X}_{t+\ell\Delta t}$ shows that it is lower than the threshold value plus the protection margin. From (22) we can deduce via Kolmogorov–Smirnov goodness-of-fit test that the residuals of the smoothed and fitted RSSI values $\hat{\epsilon}_t = X_t - \hat{X}_t$ follow

$$\hat{\epsilon}_t = X_t - \hat{X}_t \sim N(\mu_{\hat{\epsilon}}, \sigma_{\hat{\epsilon}}^2) \tag{28}$$

where $E(\hat{\epsilon}_t) = \mu_{\hat{\epsilon}}$ and $Var(\hat{\epsilon}_t) = \sigma_{\hat{\epsilon}}^2$. Hence we can write that $P(\text{false trigger}) =$

$$P(\hat{X}_{t+\ell\Delta t} \leq \bar{X} + \Delta_x^{GD} | X_{t+\ell\Delta t} > \bar{X} + \Delta_x^{GD}) = \frac{\int_{-\infty}^{\bar{X} + \Delta_x^{GD}} \left[1 - \Phi\left(\frac{\bar{X} + \Delta_x^{GD} - x - \mu_{\hat{\epsilon}}}{\sigma_{\hat{\epsilon}}}\right) \right] f_{\hat{X}}(x) dx}{1 - \int_{-\infty}^{\bar{X} + \Delta_x^{GD}} \Phi\left(\frac{\bar{X} + \Delta_x^{GD} - x - \mu_{\hat{\epsilon}}}{\sigma_{\hat{\epsilon}}}\right) f_{\hat{X}}(x) dx} \tag{29}$$

where $Z \sim N(0,1)$, $\Phi(\cdot)$ denotes the cumulative standard normal distribution function and $f_{\hat{X}}(x) = \frac{1}{\sigma_{\hat{X}_{t+\ell\Delta t}} \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{x - \mu_{\hat{X}_{t+\ell\Delta t}}}{\sigma_{\hat{X}_{t+\ell\Delta t}}}\right)^2}$ is the probability density function (pdf) of the forecasted RSSI values. On the other hand we can also define the probability of making a false non-trigger (or missed trigger) as

$$P(\hat{X}_{t+\ell\Delta t} > \bar{X} + \Delta_x^{GD} | X_{t+\ell\Delta t} \leq \bar{X} + \Delta_x^{GD}) = \frac{\int_{\bar{X} + \Delta_x^{GD}}^{\infty} \Phi\left(\frac{\bar{X} + \Delta_x^{GD} - x - \mu_{\hat{\epsilon}}}{\sigma_{\hat{\epsilon}}}\right) f_{\hat{X}}(x) dx}{\int_{-\infty}^{\infty} \Phi\left(\frac{\bar{X} + \Delta_x^{GD} - x - \mu_{\hat{\epsilon}}}{\sigma_{\hat{\epsilon}}}\right) f_{\hat{X}}(x) dx} \tag{30}$$

which is the error when the true RSSI value $X_{t+\ell\Delta t}$ is less than the enhanced threshold requirement but the forecasted RSSI value, $\hat{X}_{t+\ell\Delta t}$ shows that it is greater than the threshold value plus the protection margin. For a complete derivation of these two results, please refer to the “Appendix”.

In addition, for the forecasted RSSI values of neighbouring mesh nodes, by analogy with the probabilities of making a false trigger of the current mesh node, for each neighbouring mesh node, we define the probability of making false selection at time t of an i th node as

$$P(\hat{Y}_{t+\ell\Delta t}^{(i)} > \bar{Y}^{(i)} + \Delta_{y^{(i)}}^{CU} | Y_{t+\ell\Delta t}^{(i)} \leq \bar{Y}^{(i)} + \Delta_{y^{(i)}}^{CU}), \tag{31}$$

$i = 1, 2, \dots, M$

where $\hat{Y}_{t+\ell\Delta t}^{(i)}$ and $\bar{Y}^{(i)}$ are the i th mesh node’s forecasted RSSI value for leads $\ell \geq 1$ and its RSSI threshold value respectively. Furthermore by deducing the residuals $Y_t^{(i)} - \hat{Y}_t^{(i)}$ as

$$\hat{\epsilon}_t^{(i)} = Y_t^{(i)} - \hat{Y}_t^{(i)} \sim N\left(\mu_{\epsilon}^{(i)}, \left(\sigma_{\epsilon}^{(i)}\right)^2\right) \tag{32}$$

where $E\left(\hat{\epsilon}_t^{(i)}\right) = \mu_{\epsilon}^{(i)}$, $Var\left(\hat{\epsilon}_t^{(i)}\right) = \left(\sigma_{\epsilon}^{(i)}\right)^2$ and $Y_t^{(i)}$ is the smoothed RSSI value at time t for i th neighbouring mesh node. Hence in analogy with (31) we can write that $P(\text{false node selection}) =$

$$P\left(\hat{Y}_{t+\ell\Delta t}^{(i)} > \bar{Y}^{(i)} + \Delta_{y^{(i)}}^{CU} \mid Y_{t+\ell\Delta t}^{(i)} \leq \bar{Y}^{(i)} + \Delta_{y^{(i)}}^{CU}\right) = \frac{\int_{y = \bar{Y}^{(i)} + \Delta_{y^{(i)}}^{CU}}^{\infty} \Phi\left(\frac{\bar{Y}^{(i)} + \Delta_{y^{(i)}}^{CU} - y - \mu_{\epsilon}^{(i)}}{\sigma_{\epsilon}^{(i)}}\right) f_{\hat{Y}^{(i)}}(y) dy}{\int_{-\infty}^{\infty} \Phi\left(\frac{\bar{Y}^{(i)} + \Delta_{y^{(i)}}^{CU} - y - \mu_{\epsilon}^{(i)}}{\sigma_{\epsilon}^{(i)}}\right) f_{\hat{Y}^{(i)}}(y) dy} \tag{33}$$

where $Z \sim N(0,1)$, $\Phi(\cdot)$ denotes the cumulative standard normal distribution function and $f_{\hat{Y}^{(i)}}(y) = \frac{1}{\sigma_{\hat{Y}^{(i)}} \sqrt{2\pi}}$

$e^{-\frac{1}{2}\left(\frac{y - \mu_{\hat{Y}^{(i)}}}{\sigma_{\hat{Y}^{(i)}}}\right)^2}$ is the probability density function (pdf) of the forecasted RSSI values of the neighbouring i th mesh node with mean $\mu_{\hat{Y}^{(i)}}$ and variance $\sigma_{\hat{Y}^{(i)}}^2$. Hence we can now redefine our criterion of a handover at time t from a current mesh node to the j th mesh node as:

$$\left\{ \begin{array}{l} E\left(\hat{X}_{t+\ell\Delta t}\right) \leq \bar{X} + \Delta_x^{GD} \text{ and} \\ P\left(\hat{X}_{t+\ell\Delta t} \leq \bar{X} + \Delta_x^{GD}\right) \geq \alpha \text{ and} \\ P\left(\hat{X}_{t+\ell\Delta t} \leq \bar{X} + \Delta_x^{GD} \mid X_{t+\ell\Delta t} > \bar{X} + \Delta_x^{GD}\right) \leq \bar{\alpha} \end{array} \right\} \tag{34}$$

and

$$\left\{ \begin{array}{l} E\left(\hat{Y}_{t+\ell\Delta t}^{(j)}\right) \geq \bar{Y}^{(j)} + \Delta_{y^{(j)}}^{CU} \text{ and} \\ P\left(\hat{Y}_{t+\ell\Delta t}^{(j)} \geq \bar{Y}^{(j)} + \Delta_{y^{(j)}}^{CU}\right) \geq \beta \text{ and} \\ P\left(\hat{Y}_{t+\ell\Delta t}^{(j)} \geq \bar{Y}^{(j)} + \Delta_{y^{(j)}}^{CU} \mid \hat{Y}_{t+\ell\Delta t}^{(j)} < \bar{Y}^{(j)} + \Delta_{y^{(j)}}^{CU}\right) \leq \bar{\beta} \end{array} \right\} \tag{35}$$

where $\bar{\alpha}, \bar{\beta} \in (0, 1)$ and the index j is defined as

$$j = \left\{ i : \max\left\{P\left(\hat{Y}_{t+\ell\Delta t}^{(i)} \geq \bar{Y}^{(i)} + \Delta_{y^{(i)}}^{CU}\right), i = 1, 2, \dots, M\right\} \right\} \tag{36}$$

In order to reduce the probability of making false trigger and the probability of selecting the wrong AP to a wider margin, we can modify the above decision criteria to the following scheme:

Trigger from a current mesh node to the j th mesh node at time t when

$$\left\{ \begin{array}{l} \frac{1}{m} \sum_{i=\ell}^{\ell+m-1} E\left(\hat{X}_{t+i\Delta t}\right) \leq \bar{X} + \Delta_x^{GD} \text{ and} \\ \frac{1}{m} \sum_{i=\ell}^{\ell+m-1} P\left(\hat{X}_{t+i\Delta t} \leq \bar{X} + \Delta_x^{GD}\right) \geq \alpha \text{ and} \\ \frac{1}{m} \sum_{i=\ell}^{\ell+m-1} P\left(\hat{X}_{t+i\Delta t} \leq \bar{X} + \Delta_x^{GD} \mid X_{t+i\Delta t} > \bar{X} + \Delta_x^{GD}\right) \leq \bar{\alpha} \end{array} \right\} \tag{37}$$

and

$$\left\{ \begin{array}{l} \frac{1}{m} \sum_{i=\ell}^{\ell+m-1} E\left(\hat{Y}_{t+i\Delta t}^{(j)}\right) \geq \bar{Y}^{(j)} + \Delta_{y^{(j)}}^{CU} \text{ and} \\ \frac{1}{m} \sum_{i=\ell}^{\ell+m-1} P\left(\hat{Y}_{t+i\Delta t}^{(j)} \geq \bar{Y}^{(j)} + \Delta_{y^{(j)}}^{CU}\right) \geq \beta \text{ and} \\ \frac{1}{m} \sum_{i=\ell}^{\ell+m-1} P\left(\hat{Y}_{t+i\Delta t}^{(j)} \geq \bar{Y}^{(j)} + \Delta_{y^{(j)}}^{CU} \mid \hat{Y}_{t+i\Delta t}^{(j)} < \bar{Y}^{(j)} + \Delta_{y^{(j)}}^{CU}\right) \leq \bar{\beta} \end{array} \right\} \tag{38}$$

where $m \geq 1$, $\bar{\alpha}, \bar{\beta} \in (0, 1)$ and the index j is defined as

$$j = \left\{ i : \max\left\{P\left(\hat{Y}_{t+\ell\Delta t}^{(i)} \geq \bar{Y}^{(i)} + \Delta_{y^{(i)}}^{CU}\right), i = 1, 2, \dots, M\right\} \right\} \tag{39}$$

3.4 Trigger algorithm

Based on the analysis so far, the following is the proposed algorithm in the form of a pseudo-code:

Given the parameter values $\ell, m, \alpha, \bar{\alpha}, \beta, \bar{\beta}, \bar{X}, \bar{Y}^{(i)}, \Delta_x^{GD}, \Delta_{y^{(i)}}^{CU}, i = 1, 2, \dots, M$

- Step 1. Select a window size N from the latest smoothed RSSI values $\{X_i\}_{i=1}^N$ of the current mesh node and also $\{Y_j^{(i)}\}_{j=1}^{N^{(i)}}$ for each M neighbouring mesh nodes with their respective window size $N^{(i)}, i = 1, 2, \dots, M$
- Step 2. Extract out the jump-components from $\{X_i\}_{i=1}^N$ and $\{Y_j^{(i)}\}_{j=1}^{N^{(i)}}, i = 1, 2, \dots, M$ and estimate the OU jump diffusion process model parameters
- Step 3. Forecast smoothed RSSI values for lead time $\ell \geq \Delta t$ for all current and neighbouring mesh nodes

Step 4. **If**

$$\left\{ \begin{array}{l} \frac{1}{m} \sum_{i=\ell}^{\ell+m-1} E(\hat{X}_{t+i\Delta t}) \leq \bar{X} + \Delta_x^{GD} \text{ and} \\ \frac{1}{m} \sum_{i=\ell}^{\ell+m-1} P(\hat{X}_{t+i\Delta t} \leq \bar{X} + \Delta_x^{GD}) \geq \alpha \text{ and} \\ \frac{1}{m} \sum_{i=\ell}^{\ell+m-1} P(\hat{X}_{t+i\Delta t} \leq \bar{X} + \Delta_x^{GD} | X_{t+i\Delta t} > \bar{X} + \Delta_x^{GD}) \leq \bar{\alpha} \end{array} \right\} \text{and}$$

$$\left\{ \begin{array}{l} \frac{1}{m} \sum_{i=\ell}^{\ell+m-1} E(\hat{Y}_{t+i\Delta t}^{(j)}) \geq \bar{Y}^{(j)} + \Delta_{y^{(j)}}^{CU} \text{ and} \\ \frac{1}{m} \sum_{i=\ell}^{\ell+m-1} P(\hat{Y}_{t+i\Delta t}^{(j)} \geq \bar{Y}^{(j)} + \Delta_{y^{(j)}}^{CU}) \geq \beta \text{ and} \\ \frac{1}{m} \sum_{i=\ell}^{\ell+m-1} P(\hat{Y}_{t+i\Delta t}^{(j)} \geq \bar{Y}^{(j)} + \Delta_{y^{(j)}}^{CU} | \hat{Y}_{t+i\Delta t}^{(j)} < \bar{Y}^{(j)} + \Delta_{y^{(j)}}^{CU}) \leq \bar{\beta} \end{array} \right\} \text{for}$$

$$j = \left\{ i : \max \left\{ P(\hat{Y}_{t+i\Delta t}^{(i)} \geq \bar{Y}^{(i)} + \Delta_{y^{(i)}}^{CU}), i = 1, 2, \dots, M \right\} \right\} \text{Then}$$

Step A. Trigger a link handover to j -th mesh node.

Else If $\left\{ \begin{array}{l} \frac{1}{m} \sum_{i=\ell}^{\ell+m-1} E(\hat{X}_{t+i\Delta t}) \leq \bar{X} + \Delta_x^{GD} \text{ and} \\ \frac{1}{m} \sum_{i=\ell}^{\ell+m-1} P(\hat{X}_{t+i\Delta t} \leq \bar{X} + \Delta_x^{GD}) \geq \alpha \text{ and} \\ \frac{1}{m} \sum_{i=\ell}^{\ell+m-1} P(\hat{X}_{t+i\Delta t} \leq \bar{X} + \Delta_x^{GD} | X_{t+i\Delta t} > \bar{X} + \Delta_x^{GD}) \leq \bar{\alpha} \end{array} \right\}$

and no better neighbouring mesh nodes in the vicinity **Then**

Step B. Trigger a scan-based link handover to the “best” mesh node.

Else

Step C. Do not trigger a link handover.

Endif

Step 5. Update the latest RSSI values and return to Step 1.

4 Analysis, design and implementation of OU-LPT technique

In the previous chapter we proposed the OU jump diffusion algorithm based on modified mean-reverting diffusion process. It allows for a reliable forecast of RSSI values of local and neighbouring mesh links. This chapter contains some analyses of the proposed solution in Matlab simulation as well as in a real-time experimental testbed. The datasets adopted in our analyses represent two distinct environments namely the indoors and the outdoors. Further experimentations with different datasets may result in different levels of improvement but for initial proof of concept of our proposed OU-LPT, the existing datasets are believe to be sufficient to provide some valuable insights on what this technique may offer.

4.1 OU-LPT simulation analysis

In this section we evaluate the performance of the proposed OU-LPT technique using real RSSI data (courtesy from

Intel and Fraunhofer FOKUS) using Matlab. For the Intel dataset, RSSI values of beacon frames were measured in an indoor office environment between a laptop and an IEEE 802.11g Access Point with transmission power of 15 dBm. The laptop moved with speed of approximately 0.5 m/s around the office. The Fraunhofer dataset on the other hand, were measured outdoor (open field) between two stationary wireless mesh nodes 50 m apart. Each node was equipped with IEEE802.11g radio card with transmit power of 14 dBm.

In this analysis we strictly follow the criteria set by Intel [20] in defining the LD and LGD thresholds using its RSSI. Here the LD threshold value is set at -80 dBm and LGD threshold is set at -76 dBm which results in a protection margin, Δ_x^{GD} of 4 dB. As the RSSI values do not seem to exhibit any trends or seasonal patterns and for fast computational results, the moving average (MA) technique is the best approach as all the weights are equally distributed to the data. As for other smoothing techniques such as weighted moving average (WMA), there is a need to choose the weighting factors in an ad hoc manner or through some estimation methods and is therefore impractical for our study. Detailed analysis on various smoothing techniques though desirable, is not the focus of this paper. In the following experiments, the OU-LPT parameters used are: $N = 30$, $\ell = 5$, $\Delta_x^{GD} = 4$ dB, $m = 5$, $\alpha = 0.60$, $\bar{\alpha} = 0.10$ and smoothing window size of 10.

In Figs. 6 and 7, we display two snapshots of the triggering activities between the time 150–200 and 300–350 s respectively. From the figures we can see that the prediction mechanism is able to issue a trigger at a very early stage in preparation before the smoothed RSSI values fall below the LGD threshold. Furthermore, the presence of small number of false trigger (or false alarm) and false non-trigger (missed trigger) attest the suitability of modelling the RSSI values as a stochastic process.

In Figs. 8 and 9, we display the triggering mechanism using the same set of parameters and time intervals for the OU jump diffusion process with the exception that the prediction of future smoothed RSSI values is obtained via a linear regression (LR) method. Here for a series of N consecutive smoothed RSSI values $X_t, X_{t+\Delta t}, X_{t+2\Delta t}, \dots, X_{t+(N-1)\Delta t}$ the model is defined as

$$X_k = \beta_1 * t_k + \beta_0 + \varepsilon, \varepsilon \sim N(0, \sigma_\varepsilon^2) \tag{40}$$

where t_k is the time index for X_k ,

$$\beta_1 = \frac{\sum_{k=t}^{t+(N-1)\Delta t} (t_k - \bar{t})(X_k - \bar{X})}{\sum_{k=t}^{t+(N-1)\Delta t} (t_k - \bar{t})^2}, \beta_0 = \bar{X} - \beta_1 * \bar{t} \tag{41}$$

such that $\bar{X} = \frac{1}{N} \sum_{k=t}^{t+(N-1)\Delta t} X_k$ and $\bar{t} = \frac{1}{N} \sum_{k=t}^{t+(N-1)\Delta t} t_k$.

Fig. 6 Trigger results between 150 and 200 s using OU-LPT technique

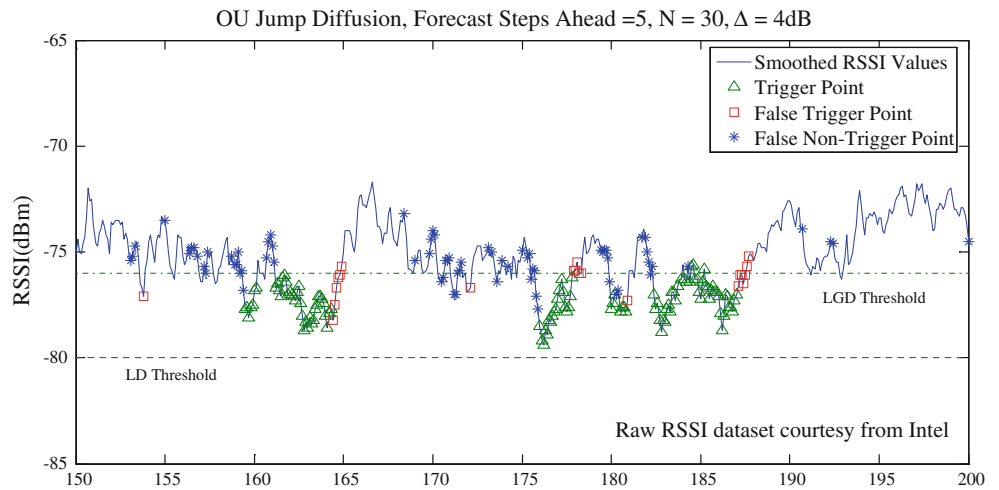


Fig. 7 Trigger results between 300 and 350 s using OU-LPT technique

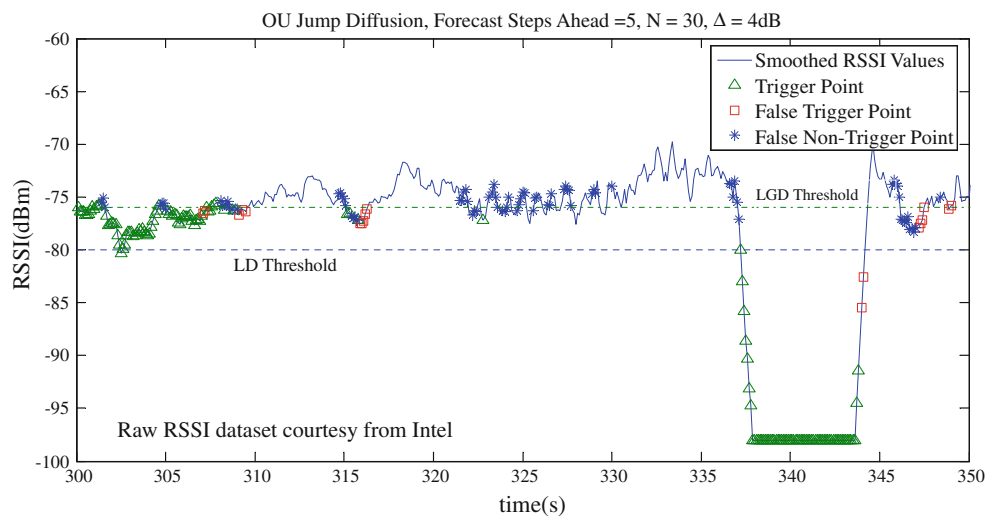
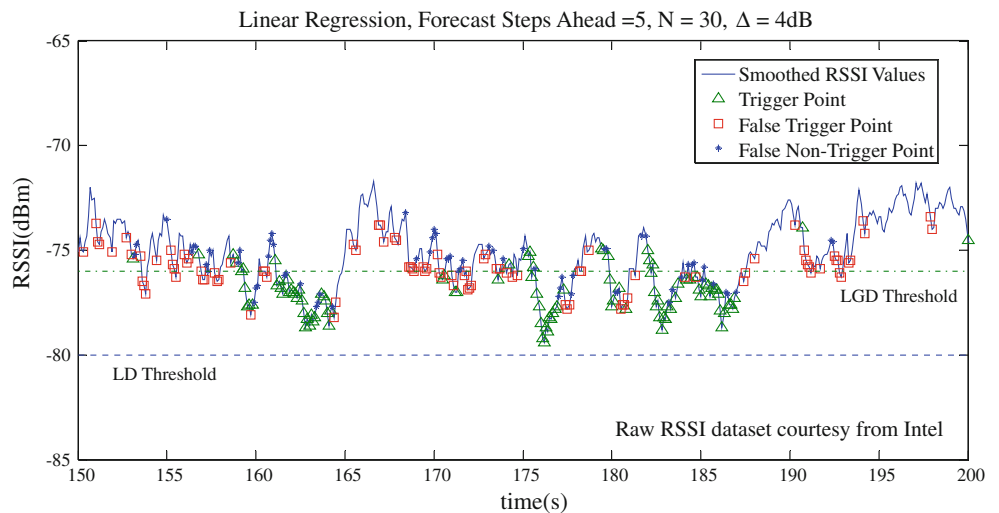


Fig. 8 Trigger results between 150 and 200 s using linear regression technique



By comparing Figs. 6, 7, 8 and 9 we can see by using the linear regression approach there is a higher likelihood that a false trigger would occur as compared with the

approach taken by the proposed OU-LPT technique. In this paper we only compare OU-LPT with LR as both models are linear in construction and hence we are assessing

Fig. 9 Trigger results between 300 and 350 s using linear regression technique

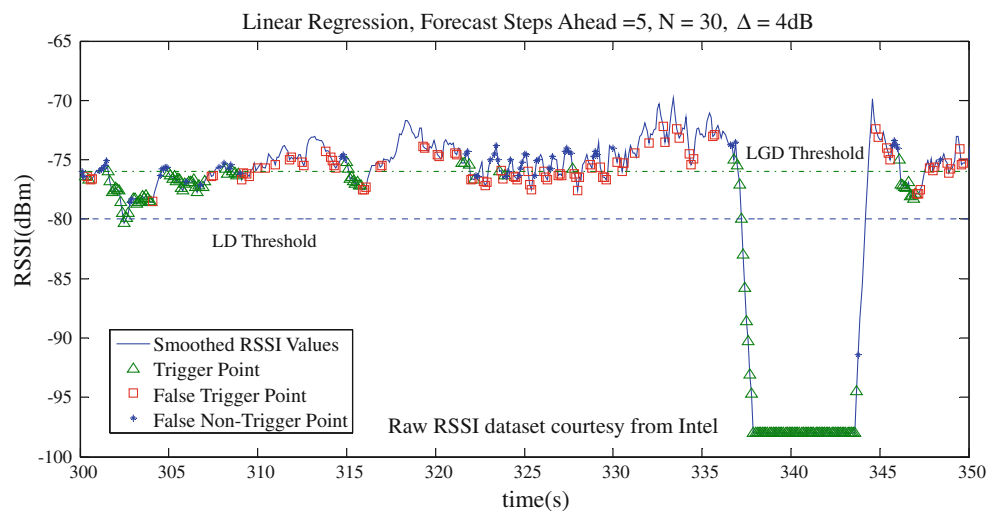


Table 2 Trigger results for Intel dataset of OU-LPT and linear regression techniques

Description of Trigger	OU-LPT (%)	LR (%)	Improvement (%)
Triggers	24.20	36.49	-12.29
False triggers	7.63	38.10	-30.47
Non-triggers	75.80	63.51	+12.29
False non-triggers	10.46	9.90	+0.56

like-for-like. Take note that the proposed OU-LPT is based on stochastic process modelling of the velocity of the random movements of RSSI values whilst the LR only looks into the relationship between explanatory and response variables. On the other hand time series models are not considered in this study as they are too computationally intensive such as there is a need to perform stationary test of the data, model identification, parameters estimation as well as diagnostic checking before one can fully use it. Therefore due to time constraints in the triggering process we have to exclude this technique. Furthermore time series models are not as practical as our OU-LPT technique from the implementation point of view.

Table 2 shows the comparison of trigger statistics between OU-LPT and linear regression method. From the table we can see that using our proposed method there is a significant improvement in reducing the rate of committing false trigger (7.63 % out of 24.20 % of trigger occurrences) as compared with the conventional linear regression method (38.10 % out of 36.49 % of trigger occurrences) which is a brute strength method without taking into account of modelling fat-tails distribution. However the percentage of committing a false non-trigger for either both methods are quite comparable (10.46 and 9.90 %). We also notice that, the percentage of false non-trigger (missed trigger) = 10.46 % whilst the percentage of false trigger

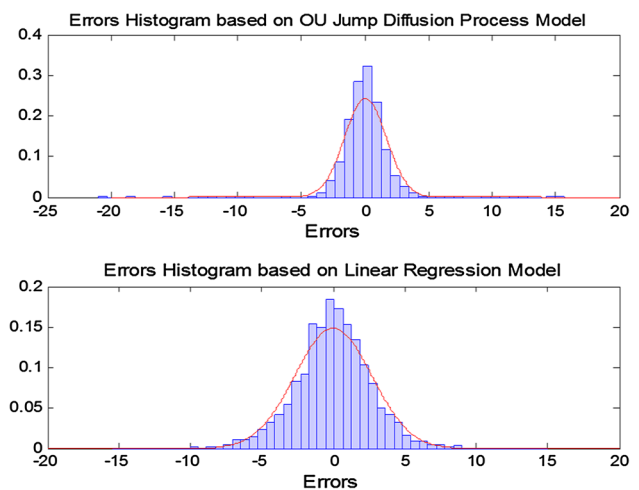


Fig. 10 Histograms of errors with fitted normal densities (shown overlaid) for OU-LPT and linear regression techniques

Table 3 Link state prediction result or trigger

Output/prediction	Description	Event type
LINK_DOWN	Link completely down	State change
LINK_GOING_DOWN	High probability of the link losing its connection status	Predictive
LINK_UP	The Link is above the threshold value	State change
LINK_GOING_UP	The probability of the link recovering its signal is high	Predictive

(false alarm) = 7.63 %. The discrepancy can be due to the high volatility of the signal as well as the selection of the protection margin of 4 dB in which most of the RSSI values reside near -76 dB. This observation is therefore

specific to Intel’s dataset. In Fig. 10 we show the error analysis of both methods and from the histogram plots we can deduce that the errors generated from both methods are

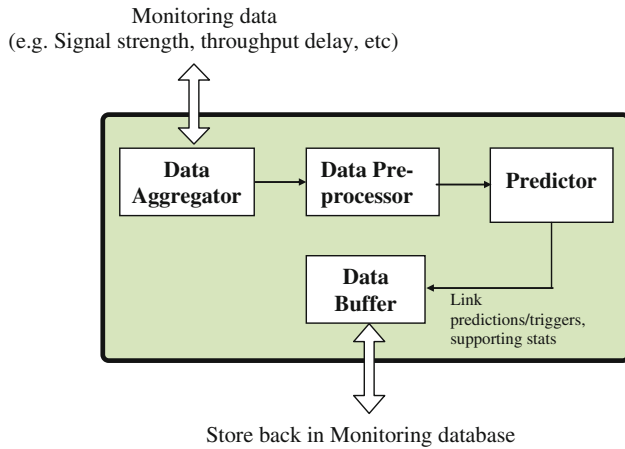
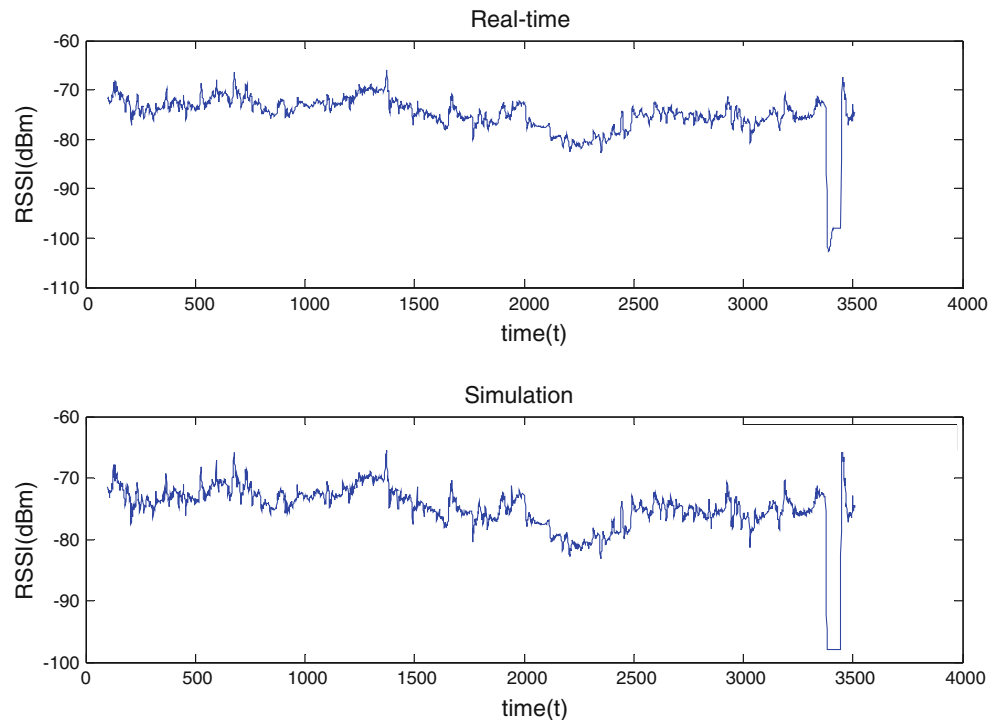


Fig. 11 General architecture of the OU-LPT Module

Table 4 Default settings for OU-LPT operation

<i>General OU-LPT</i>	
Data sample interval/step size	100 ms
Moving average window size	10
<i>Jump diffusion algorithm</i>	
Prediction Window size, N	30
Prediction steps (or look ahead time)	5 steps (500 ms)
Protection margin	4 db
LD threshold	−80 dBm

Fig. 12 Comparison between real-time and Matlab prediction on RSS values (Intel data)



approximately normal distributed. However by comparing the two approaches we can see that the errors from OU-LPT tend to have a smaller standard deviation, and hence the errors are less dispersed.

Although both methods have comparable lead time which is the time difference between the first successful trigger until the signal strength goes below the LD threshold, by reducing the chances of making a false trigger or missed trigger, the proposed OU-LPT technique is by far a more reliable method than linear regression.

4.2 OU-LPT module design and real-time implementation

Fast variations of radio channel characteristics entail the need of smoothing mechanism introduction to deal with raw data measurements, as well as to avoid incorrect decisions based on temporary measured values of parameters. To overcome this problem, a double level averaging process has been implemented within a measurement modules and monitoring aggregation module as well. Additional long-term statistics repository has been created to provide feedback mechanism for routing with link stability description. Also link related events are predicted and reported by link triggering and prediction module. It determines state change and predictive events as shown in Table 3 as defined in IEEE802.21 [21]. In mesh environment, the reliability of wireless backhaul links is extremely critical as any link disruption may affect more than one

node. For that reason, such predictive triggers are particularly important to ensure carrier-grade performance.

Figure 11 presents the general architecture of the OU-LPT module. The Data Aggregator submodule is responsible for pulling the required raw data. The data, which is generally retrieved on a fixed interval, can be RSSI, throughput or delay depending on the usage requirement. After gathering of the required data, these values would then be passed on to the Data Pre-processor submodule. The Data Pre-processor submodule is generally responsible to prepare the data before passing on to the prediction submodule. The tasks include reformatting, synchronization with real-time clock, re-sampling and smoothing as required by the predictor submodule. The smoothing process aims to reduce the fluctuation in the raw signal values

and also helps to convert the time series data into a data set with fewer fluctuations. This will help prevent unnecessary triggers later on. As the name implies, predictor submodule analyzes the time series data and predict the future state of the link i.e. LGD. Alternatively, other conventional prediction algorithms such as Linear Regression, Lagrange or Newton extrapolation, etc., can be used. The link state prediction results, together with computed data such as trigger and errors probabilities are subsequently stored in Data Buffer. This prediction data can also be stored back inside some repository for further processing.

The OU-LPT module has been implemented using C and runs on a relatively slow Soekris net5501 500 MHz 586 class embedded system board together with other modules defined in the CARMEN project for the mesh node architecture (resource aware routing, admission control, spectrum management, monitoring, self-configuration and support for mobile users). The OU-LPT module comprises of around one thousand over lines of code and utilizing only C native libraries. When comparing with Matlab implementation, the real-time implementation posts a greater set of challenges as there are different options to implement the same math function. The decision on which approach to be adopted will have an impact on the accuracy and computation time. In our implementation, the average time required to generate a prediction using the above-mentioned board with 512 Mbyte DDR-SDRAM running UBUNTU LTE 8.04 operating system is less than 28 ms. This is much lower than 100 ms, which is the time interval in order for a prediction to be useful. The

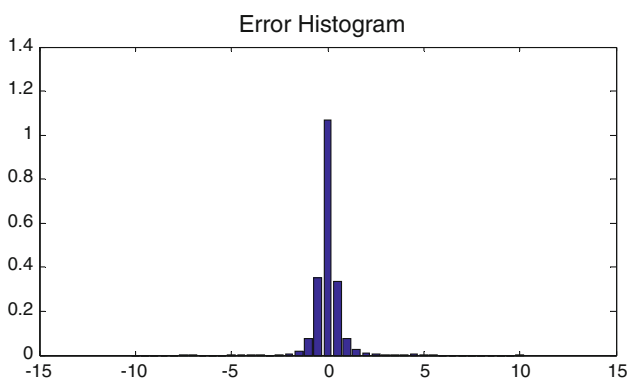


Fig. 13 Error histogram comparing Matlab simulation and real-time OU-LPT implementation (Intel dataset)

Fig. 14 Scatter plot comparing Matlab and real-time OU-LPT implementation (Intel dataset)

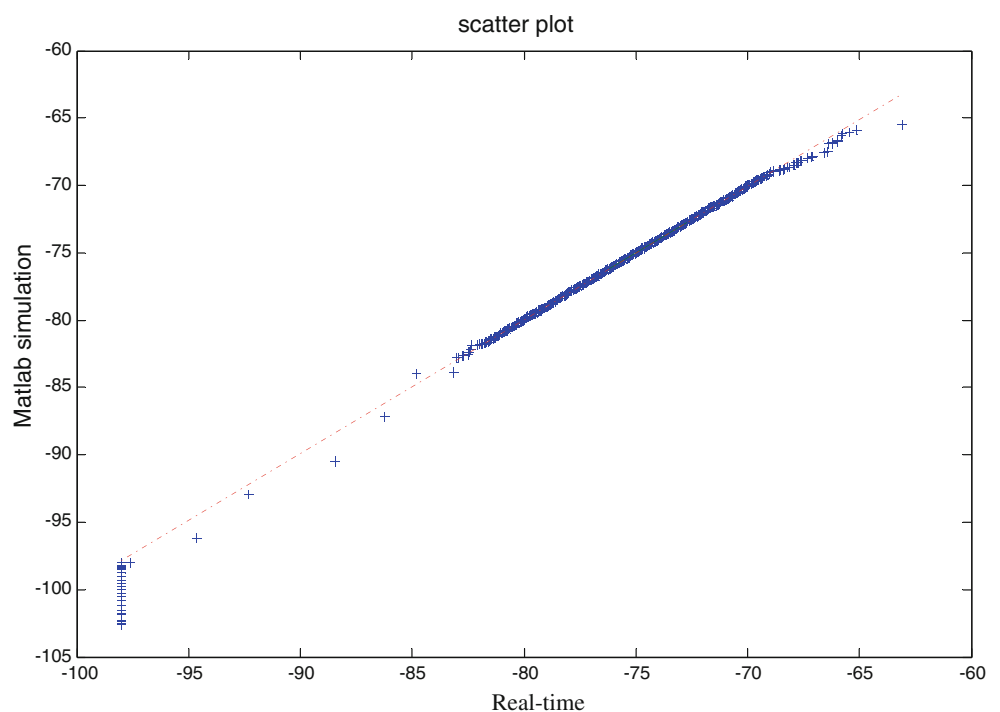


Fig. 15 Comparison between real-time and Matlab prediction on RSSI values (FHG dataset)

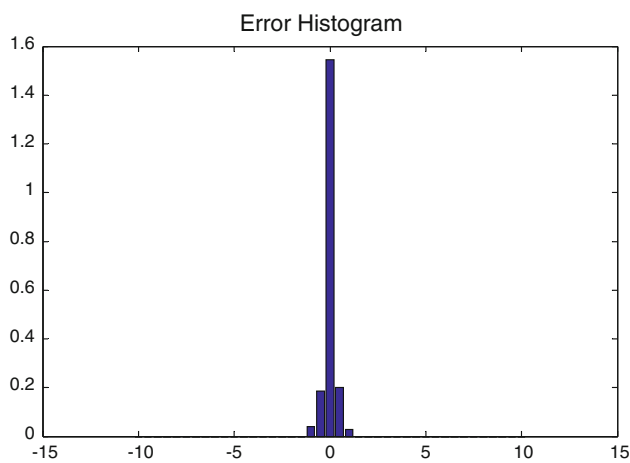
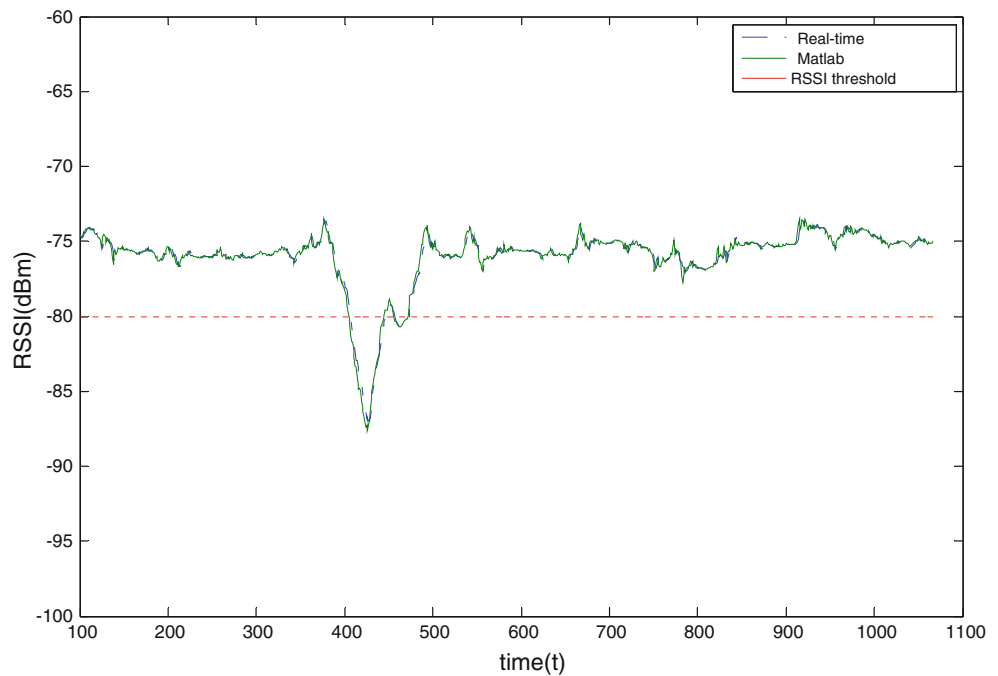


Fig. 16 Error histogram comparing Matlab and real-time OU-LPT implementation (FHG dataset)

prediction process also consumes around 6.9 MIPS of CPU usage. Although the OU-LPT module is able to do real-time capture, we have instructed it to read the same pre-recorded datasets (from Intel and Fraunhofer) as if it is acquiring the signal from the WLAN card in real-time to enable direct comparison between the performance of real-time implementation and Matlab implementation. Table 4 summarizes the parameters used:

Figure 12 compares the predicted RSS values of real-time and Matlab simulation using the Intel dataset. As observed, the predicted values are relatively close. On average however, Matlab produces less bursty values across the whole data set. This can be observed around

time 3,400 s, where the real-time result shows higher variance compared to Matlab result.

Figure 13 shows the error histogram comparing Matlab and real-time OU-LPT technique. The result shows that the predictions generated by the real-time OU-LPT are only slightly deviated from the predictions generated by Matlab. When investigating across time, the predicted values are basically the same except at certain points when differences appear. The errors mainly occur during the drastic change of signal strength at around 3400 s (Fig. 12). This is confirmed by fat-tail effect at the left hand side of the scatter plot in Fig. 14. It is also observed that the errors generally occur at lower RSS values (< -98 dBm) therefore may not affect the accuracy of triggers. The discrepancies between Matlab and real-time C implementation are believed to be caused by different approaches in implementing certain mathematical functions such as the integration.

Figure 15 shows the analysis using another dataset (contributed by Fraunhofer FOKUS). This data does not have drastic drop of signal strength and even there is, the drop is very gradual and hence easy to be predicted.

It can be observed from Figs. 16 and 17 that when analyzing the dataset provided by FHG, the error between the Matlab and real-time is relatively small. The issue of fat tail does not exist in this case. This is due to the less drastic trend in signal fluctuation. The above evaluation shows that the OU-LPT technique can be implemented with relatively small computation overhead and complexity.

The OU-LPT graphical visualizer has also been developed using GTK toolbox (<http://www.gtk.org>) which is also part of the GNU project. It is a cross-platform widget

Fig. 17 Scatter plot comparing Matlab and real-time OU-LPT implementation (FHG dataset)

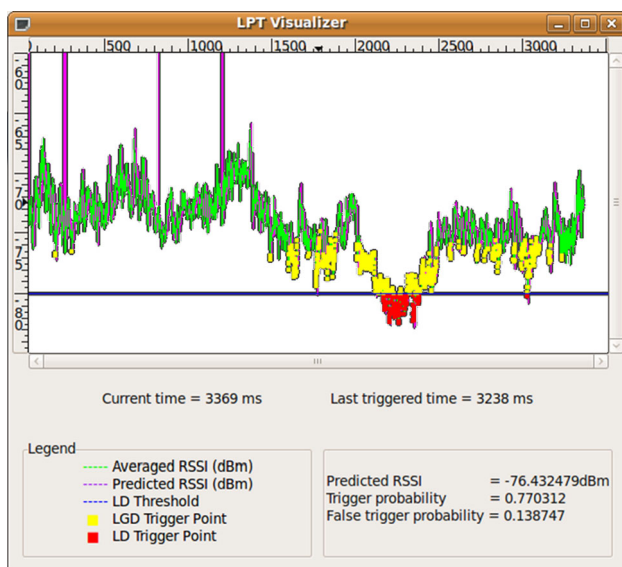
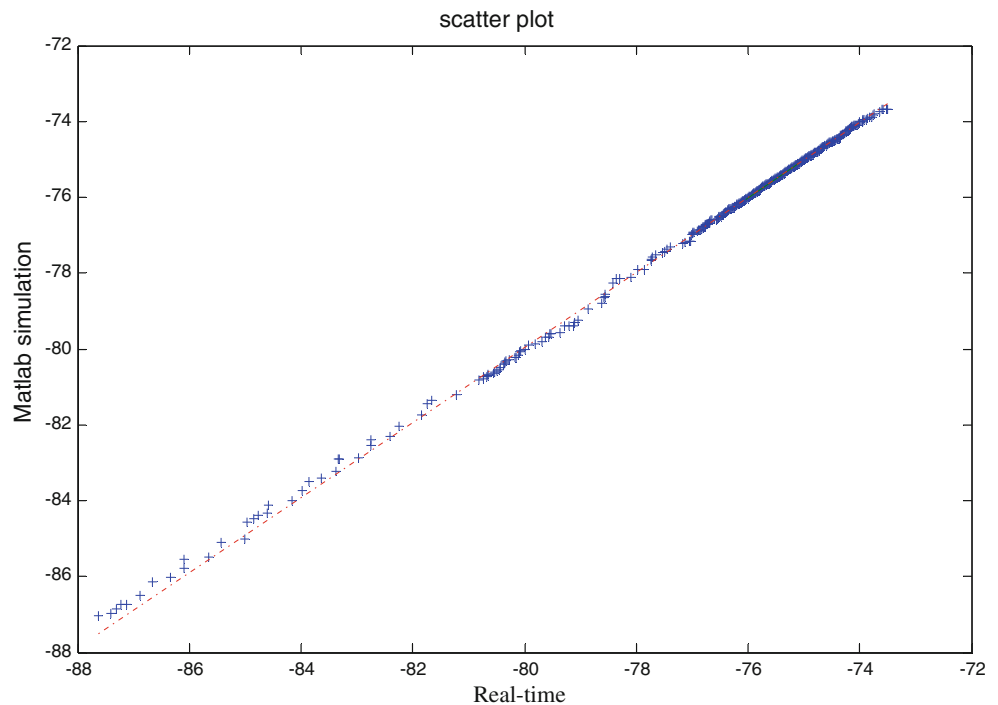


Fig. 18 Time series line plots and prediction probabilities on OU-LPT visualizer

toolkit used to develop GUI. Figure 18 gives a snapshot of the OU-LPT visualizer.

The visualizer shows the current smooth signal and also the predicted signal. The yellow markers indicate the LGD trigger events while the red markers indicate the LD events. The dark blue line represents the LD threshold. Other statistics such as trigger probability and false trigger probability of each trigger can be computed and shown in real-time.

5 Conclusions

Monitoring system is an integral part of every wireless mesh network. It provides to other modules accurate and timely information regarding the status of a network as well as to predict the quality of the wireless link. The results of prediction are used to reconfigure the network in advance to avoid service disruption. This paper proposes a novel link prediction and triggering technique based on a modified mean-reverting diffusion process. The analysis shows that the proposed OU-LPT method can significantly enhance the reliability of wireless links which is particularly critical in wireless mesh environment. A significant improvement has been observed in reducing the rate of committing false trigger (from 38.1 to 7.63 % out of total trigger occurrences) as compared with the conventional linear regression method. The proposed method also incurs a very small percentage of false trigger when compared to the conventional linear regression method. On top of that when comparing the errors, OU-LPT experiences a smaller standard deviation implying that the errors are less dispersed. The link-up scenario is not addressed in this paper because it generally operates in the direct opposite manner as link-down scenario. The prediction on link-up however can be used for early preparation of link to its normal operation. The proposed OU-LPT algorithm has also been successfully implemented and evaluated using a real-time embedded system board. Overall the OU-LPT technique is found to be promising and it offers a new direction on how wireless link prediction, triggering and switching process can be conducted in the future.

Acknowledgments This work was partially funded by the European Commission within the 7th Framework Program in the context of the ICT project Carrier-Grade Mesh Networks (CARMEN) (Grant Agreement No. 214994). The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the CARMEN project or the European Commission. The authors would also like to thank Intel and Fraunhofer FOKUS for contributing the datasets.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

Appendix

The following are the steps to derive the probabilities of making a false trigger (or false alarm) and making a false non-trigger (or missed trigger) for the link handover process.

$$\begin{aligned}
 P(\text{false trigger}) &= P(\hat{X}_{t+\ell\Delta t} \leq \bar{X} + \Delta_x^{GD} | X_{t+\ell\Delta t} > \bar{X} + \Delta_x^{GD}) \\
 &= \frac{P(\hat{X}_{t+\ell\Delta t} \leq \bar{X} + \Delta_x^{GD}, X_{t+\ell\Delta t} > \bar{X} + \Delta_x^{GD})}{P(X_{t+\ell\Delta t} > \bar{X} + \Delta_x^{GD})} \\
 &= \frac{P(X_{t+\ell\Delta t} > \bar{X} + \Delta_x^{GD}, \hat{X}_{t+\ell\Delta t} \leq \bar{X} + \Delta_x^{GD})}{P(X_{t+\ell\Delta t} > \bar{X} + \Delta_x^{GD})} \\
 &= \frac{P\left(Z > \frac{\bar{X} + \Delta_x^{GD} - \hat{X}_{t+\ell\Delta t} - \mu_{\hat{x}}}{\sigma_{\hat{x}}}, \hat{X}_{t+\ell\Delta t} \leq \bar{X} + \Delta_x^{GD}\right)}{P\left(Z > \frac{\bar{X} + \Delta_x^{GD} - \hat{X}_{t+\ell\Delta t} - \mu_{\hat{x}}}{\sigma_{\hat{x}}}\right)} \\
 &= \frac{\int_{-\infty}^{\bar{X} + \Delta_x^{GD}} \left[1 - P\left(Z \leq \frac{\bar{X} + \Delta_x^{GD} - \hat{X} - \mu_{\hat{x}}}{\sigma_{\hat{x}}} \middle| \hat{X} = x\right)\right] f_{\hat{X}}(x) dx}{1 - \int_{-\infty}^{\infty} P\left(Z \leq \frac{\bar{X} + \Delta_x^{GD} - \hat{X} - \mu_{\hat{x}}}{\sigma_{\hat{x}}} \middle| \hat{X} = x\right) f_{\hat{X}}(x) dx} \\
 &= \frac{\int_{x=\bar{X} + \Delta_x^{GD}}^{\infty} \left[1 - \Phi\left(\frac{\bar{X} + \Delta_x^{GD} - x - \mu_{\hat{x}}}{\sigma_{\hat{x}}}\right)\right] f_{\hat{X}}(x) dx}{1 - \int_{-\infty}^{\infty} \Phi\left(\frac{\bar{X} + \Delta_x^{GD} - x - \mu_{\hat{x}}}{\sigma_{\hat{x}}}\right) f_{\hat{X}}(x) dx}. \quad (42)
 \end{aligned}$$

$$\begin{aligned}
 P(\text{false non-trigger}) &= P(\hat{X}_{t+\ell\Delta t} > \bar{X} + \Delta_x^{GD} | X_{t+\ell\Delta t} \leq \bar{X} + \Delta_x^{GD}) \\
 &= \frac{P(\hat{X}_{t+\ell\Delta t} > \bar{X} + \Delta_x^{GD}, X_{t+\ell\Delta t} \leq \bar{X} + \Delta_x^{GD})}{P(X_{t+\ell\Delta t} \leq \bar{X} + \Delta_x^{GD})} \\
 &= \frac{P(X_{t+\ell\Delta t} \leq \bar{X} + \Delta_x^{GD}, \hat{X}_{t+\ell\Delta t} > \bar{X} + \Delta_x^{GD})}{P(X_{t+\ell\Delta t} \leq \bar{X} + \Delta_x^{GD})} \\
 &= \frac{P\left(Z \leq \frac{\bar{X} + \Delta_x^{GD} - \hat{X}_{t+\ell\Delta t} - \mu_{\hat{x}}}{\sigma_{\hat{x}}}, \hat{X}_{t+\ell\Delta t} > \bar{X} + \Delta_x^{GD}\right)}{P\left(Z \leq \frac{\bar{X} + \Delta_x^{GD} - \hat{X}_{t+\ell\Delta t} - \mu_{\hat{x}}}{\sigma_{\hat{x}}}\right)} \\
 &= \frac{\int_{\bar{X} + \Delta_x^{GD}}^{\infty} P\left(Z \leq \frac{\bar{X} + \Delta_x^{GD} - \hat{X} - \mu_{\hat{x}}}{\sigma_{\hat{x}}} \middle| \hat{X} = x\right) f_{\hat{X}}(x) dx}{\int_{-\infty}^{\infty} P\left(Z \leq \frac{\bar{X} + \Delta_x^{GD} - \hat{X} - \mu_{\hat{x}}}{\sigma_{\hat{x}}} \middle| \hat{X} = x\right) f_{\hat{X}}(x) dx} \\
 &= \frac{\int_{x=\bar{X} + \Delta_x^{GD}}^{\infty} \Phi\left(\frac{\bar{X} + \Delta_x^{GD} - x - \mu_{\hat{x}}}{\sigma_{\hat{x}}}\right) f_{\hat{X}}(x) dx}{\int_{-\infty}^{\infty} \Phi\left(\frac{\bar{X} + \Delta_x^{GD} - x - \mu_{\hat{x}}}{\sigma_{\hat{x}}}\right) f_{\hat{X}}(x) dx}. \quad (43)
 \end{aligned}$$

References

- Azcorra, A. et al. (2009). Supporting carrier grade services over wireless mesh networks: The approach of the European FP-7 STREP CARMEN. *IEEE Communications Magazine*, 47(4), 14–16
- Banchs, A. et al. (2008). “CARMEN: Delivering carrier grade services over wireless mesh networks”, IEEE PIMRC 2008—Social and mesh networking workshop, Cannes, French Riviera, France, September 15–18, 2008.
- Simsek, B., Chen, J. Y., Chieng, D., Natkaniec, M., & Łoziak, K. (2009). “Self configuration architecture for carrier grade mesh network (CARMEN)”, 18th ICT mobile & wireless communications summit, Santander, Spain, June 10–12, 2009.
- Natkaniec, M., Kosek-Szott, K., Szott, S., Gozdecki, J., Głowacz, A., Sargento, S. (2011) Supporting QoS in integrated ad-hoc networks. *Wireless Personal Communications*, 56(2), 183–206.
- Szott, S, Natkaniec, M., & Banchs, A. (2009). Impact of misbehaviour on QoS in wireless mesh networks. IFIP networking, May 11–15, 2009.
- Łoziak, K., Natkaniec, M., Gozdecki, J., Chin, E., Chieng, D., & Teh, V. (2010). Monitoring system for carrier grade MESH networks. Future network and mobile summit 2010, Florence, June 16–18, 2010.
- Garbaczewski, P., & Olkiewicz, R. (2000). Ornstein–Uhlenbeck–Cauchy process. *Journal of Mathematical Physics*, 41, 6843–6860.
- EU FP7 ICT Carrier Grade Mesh Networks. <http://www.ict-carmen.eu>.
- Kim, K., & Shin, K. G. (2006). *On accurate measurement of link quality in multi-hop wireless mesh networks: Proceedings of ACM MobiCom’06*, LA, USA, September 2006.
- LaCurtis, K., & Balakrishnan, H. (2010). *Measurement and analysis of real-world 802.11 mesh networks: Proceedings of the 10th annual conference on internet measurement (IMC ’10)*, Melbourne, Australia, November 1–3, 2010.
- Farkas, K., Hossmann, T., Legendre, F., Plattner, B., & Das, S. K. (2008). Link quality prediction in mesh networks. *Computer Communications*, 31(8), 1497–1512.
- Long, X., & Sikdar, B. (2008) *A real-time algorithm for long range signal strength prediction in wireless networks: Proceedings of WCNC’2008*, March 31–April 3, Las Vegas, Nevada, USA, 2008.
- Aguayo, D., Bicket, J., Biswas S., Judd, G. & Morris, R. (2004) Link-level measurements from an 802.11b mesh network. In ACM SIGCOMM 2004, August 2004.
- <http://madwifi-project.org/>.
- IEEE Std 802.11-2007, IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless lan medium access control (MAC) and physical layer (PHY) specifications, June 2007.
- Campista, M. E. M., et al. (2008). Routing metrics and protocols for wireless mesh networks. *IEEE Network*, 22(1), 6–12.
- Byrd, R. H., Lu, P., & Nocedal, J. (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific and Statistical Computing*, 16(5), 1190–1208.
- Cont, R. & Tankov, P. (2003). *Financial modelling with jump processes*. London: Chapman and Hall.
- Ito, K. (1951). On stochastic differential equations. *American Mathematical Society*, 4, 1–51.
- Liu, H., Maciocco, C., Kesavan, V., & Low, A. L. Y. (2008) *A smart triggering scheme to reduce service interruption during heterogeneous handovers: Proceedings of IEEE international*

conference on dependable systems and networks, (DSN'08), 2008.

21. IEEE Std 802.21-2009, IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 21: Media Independent Handover Services, January 2009.

Author Biographies



Eric Chin is a Senior Researcher at BT Innovate and Design and since 2007 has joined the Intelligent Systems Research Centre in the United Kingdom where his research focused on revenue, risk and radio resource management issues. Prior to that he was attached to BT Asian Research Centre in Malaysia principally investigating wireless technologies such as power control, mobile ad hoc networks and heterogeneous networks hand-over issues. Eric Chin holds a

Ph.D. in Numerical Optimisation from University of Dundee, Scotland and also master of science degrees in Applied Mathematics and Applied Statistics from University of Malaya, Malaysia and Keble College Oxford respectively.



David Chieng joined BT Malaysian Research Centre as a technical group leader in broadband wireless research group in 2006. He led a work package in EU ICT FP-7 Carrier Grade Wireless Mesh Network (CARMEN) project. Prior to BT, he was the principal researcher in project BONDS (Brokering Open Network Differentiated Services), a collaborative project between Queen's University of Belfast, Ulster University and Fujitsu Telecommunications Europe

Ltd. From 2002 to 2003, he worked as a product specialist (VoIP solutions) for a regional telecommunication software and system integration company. From 2003 to 2006, he was a lecturer, then senior lecturer in Faculty of Engineering, Multimedia University, Malaysia and was leading two joint research projects with Panasonic R&D Centre, Malaysia, in the areas of Computer Telephony Integration and Ubiquitous Computing. David received his M.Sc. and Ph.D. from the School of Electrical and Electronic Engineering, Queen's University of Belfast, Northern Ireland, UK, in 1998 and 2002 respectively. He has authored and co-authored more than 30 publications in international journals, conferences and book chapters. He has also filed 7 patents in the area of wireless and computing systems.



Victor Teh received his B.Eng in Electronic Engineering (Majoring in Microwave and Communication) from Multimedia University, Malaysia. He was an intern with BT Malaysian Research Centre between Jun and September 2009 working on wireless link prediction algorithm development. He continued the same research as his final year undergraduate project from September 2009 to April 2010.



Marek Natkaniec received the M.Sc. and Ph.D. degrees in telecommunications from the Faculty of Electrical Engineering, AGH University of Science and Technology, Cracow, Poland in 1997 and 2002, respectively. In 1997 he joined AGH University of Science and Technology. Now, he works as an assistant professor at the Department of Telecommunications. His general research interests are in wireless networks. Particular topics include wireless LANs,

designing of protocols, modeling and performance evaluation of communication networks, quality of service, and cooperation of networks. He has served as a Technical Program Committee member in many international conferences. He is also a reviewer to international journals and conferences. He has/had actively participated in European projects: MOCOMTEL, PRO-ACCESS, DAIDALOS I and II, CONTENT, CARMEN, MEDUSA, HECTOR, FLAVIA, PROACTIVE as well as grants supported by the Ministry of Science and Higher Education. He is involved in standardization activities for ETSI. He serves as an expert and a consultant to telecom operators in the area of wireless networks. Marek Natkaniec co-authored 8 books and over 100 research papers. He is a senior member of IEEE.



Krzysztof Loziak received the M.Sc. degree in telecommunications from the Faculty of Electrical Engineering, AGH University of Science and Technology, Cracow, Poland in 1999. In 2001 he has joined the Department of Telecommunications of the AGH University of Science and Technology and now works there as a lecturer. His general research interests cover the wireless communication networks, protocols design and deployment, routing optimi-

sation protocols and networks performance evaluation and optimisation. He had/had been actively participated in several European projects, such as: MOCOMTEL, JOINT, PRO-ACCESS, DESYME,

DAIDALOS I, DAIDALOS II, CARMEN, MEDUSA, FLAVIA, PROACTIVE as well as national grants supported by the Polish Ministry of Science and Higher Education. Krzysztof Loziak co-authored three books and over 50 research papers presented on conferences or published in magazines.



Janusz Gozdecki received his M.Sc. and Ph.D. degrees in telecommunications from the AGH University of Science and Technology, Krakow, Poland in 1995 and 2010, respectively. In 1995 he joined AGH-UST as a researcher. Now, he works as an assistant professor at the Department of Telecommunications. His general research interests are in wireless networks and quality of service in IP networks. He is a reviewer for international journals and conferences. He has actively partici-

parted in several European projects (BTI, MOBYDICK, DAIDALOS I,

DAIDALOS II, CARMEN, MEDUSA, HECTOR, FLAVIA, PROACTIVE) as well as grants supported by the Ministry of Science and Higher Education. Janusz Gozdecki is the co-author of 6 books and over 60 research papers.