A Handover Triggering Algorithm for Managing Mobility in WSNs

Jianjun Wen and Waltenegus Dargie

Chair for Computer Networks, Faculty of Computer Science, Technical University of Dresden, 01062 Dresden, Germany
Email:{jianjun.wen, waltenegus.dargie}@tu-dresden.de

***Abstract*—Wireless sensor networks are useful for a large number of healthcare applications which require mobile nodes. Most existing applications rely on body area networks (BAN) which require the presence of additional devices, such as mobile phones, to transfer data from the BAN to a remote base station or a server. In this paper we propose to extend BAN with personal area networks (PAN) so that enhanced mobility and seamless collection of data can be possible. In order to improve the reliability of this merge and support a high goodput, we also propose a seamless handover mechanism which enables mobile transmitters to discover and transfer communication to reliable relay nodes when the quality of an existing link deteriorates. In this paper we shall report how we implemented our scheme for TinyOS and TelosB platforms and compared it with four other competitive schemes.**

***Index Terms*—BAN, Handover, mobility management, MAC, PAN, wireless sensor network**

I. Introduction

Wireless sensor networks are useful for a large number of healthcare applications which require mobile nodes. For instance, the wireless motility capsule integrating pH, pressure, and temperature sensors for the diagnosis of gastroparesis has officially been approved by the US drug and food administration since 2006; it has produced promising results and may replace existing invasive and painful procedures (such as endoscopy) [12]. Most existing or proposed healthcare applications rely on Body Area Networks (BAN) and are often self-contained. In a BAN, nodes transfer sensed data to a mobile phone or a laptop computer which is carried by the user or is placed nearby. In some applications, individual nodes temporarily store the data they sensed locally, which are then offloaded, either manually or automatically, to a base station whenever the user happens to be at a close proximity. The advantage of the first strategy is that live and steady monitoring can be supported. One of its disadvantages is that the user is forced to always carry an additional device (mobile phone or a laptop) or make do of restricted mobility. The advantage of the second is that the user can enjoy unrestricted mobility but the applications have to be delay tolerant. Moreover, individual nodes should have sufficient storage.

The scope and usefulness of proposed healthcare applications can greatly be enhanced if the BAN they employ are augmented by Personal Area networks (PAN). In places such as home or rehab centres, additional and stationary nodes can be strategically placed, so that the BAN can interact with them to transfer data to a remote base station where they can be available for an expert or advanced data processing. There are two formidable challenges, however. Firstly, most healthcare applications require high goodput, low latency, and low jitter, to ensure that the sensed data are reliable (for example, in order to determine whether measurements are correlated). Secondly, it is difficult to maintain a reliable link between a mobile node and a stationary node, because the quality of a wireless link, in addition to distance, is also strongly dependent on mobility. In order to highlight the second challenge, we refer to Fig. 1. We placed two stationary relay nodes in the foyer of our faculty, separated from one another by a distance of $30\,\mathrm{m}$. There were no objects between these nodes to obstruct communication. A mobile robot carrying a transmitting node moves at a speed of approximately $0.13\,\mathrm{m/s}$ from one of the nodes to the other in a straight line whilst the transmitter continuously transmitted packets to both nodes simultaneously (4800 packets in all). We plotted the received signal strength indicator (RSSI) of ACK packets as a function of distance.

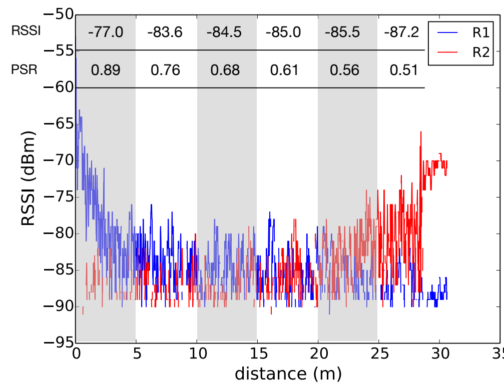


Fig. 1: The fluctuation of link quality during mobility and the associated packet success rate in different regions.

As can be seen in Fig. 1, the signal strength of the received packets fluctuated considerably for both receiving nodes, regardless of the relative distance of the robot from the receivers. Another interesting aspect we observed in this experiment was that some packets were lost even though the received signal strength of neighbouring packets indicated that they should have been successfully received. By the same token, some packets were successfully received even though their RSSI was too small. In order to deal with these challenges (to increase the reliability and throughput of mobile, wireless links), we propose a seamless handover. Unlike the handover strategies applicable for cellular networks, however, our proposed strategy does not rely on resource-rich base stations which determine when and how a mobile node should transfer communication. Instead, the mobile node itself, by examining the fluctuation of the RSSI values of incoming acknowledgement packets and the packet success rate, seamlessly transfers a communication from one relay node to another without the need to first disconnect an existing communication. For this reason, it is vital for a mobile node to (1) determine whether a fluctuation in link quality eventually results in a disconnection, (2) foresee potential disconnection well ahead of time and establish an alternative link before the disconnection occurs, and (3) seamlessly transfer communication to the new link. In this paper, we address (1) and (2).

The remaining part of this paper is organized as follows: In Section II, we review related work and position our own work. In Section III, we propose a Kalman filter based handover trigger algorithm. In Section IV, we describe implementation details and integration with existing MAC protocol. In Section V, we provide quantitative results and evaluate their implication. Finally, in Section VI we give concluding remarks and outline future work.

II. Related Work

Wireless sensor networks which support mobile nodes require mobility management protocols. The task of these protocols is to continuously evaluate the quality of a link and transfer communication to a more reliable link when the quality or the packet loss rate of an existing link deteriorates beyond a certain level (this level is defined by the application according to its quality of service requirement such as the acceptable packet loss rate).

Most existing mobility management protocols are extensions of existing asynchronous, contention-based MAC protocols. Some of these are MRI-MAC [4], MA-MAC [16], MX-MAC [3], X-Machiavel [8], and ME-ContikiMAC [11]. The first protocol is receiver-initiated, whilst the others are sender-initiated. Since all these protocols are asynchronous (they enable nodes to sleep and wakeup asynchronously), they broadcast either short preamble or beacon to announce their desire or readiness for communication. In [4], the authors assume that there is a one-to-one mapping between the RSSI values and the distance separating the transmitter and the receiver. The mobile node first transmits n packets to estimate its relative distance from the receiver and if the relative distance is beyond a predefined threshold (which implies that subsequent packets may not be transmitted successfully), it triggers a handover immediately. In [16] and [3], both protocols support a seamless handover by enabling a node to search for an alternative relay node whilst communicating with an established link. The first protocol focuses on defining two RSSI thresholds to trigger a handover, whereas the latter focuses on estimating the actual RSSI fluctuation using a least-mean-square filter (LMS). X-Machiavel [8] and ME-ContikiMAC [11] use specific control packets rather than data packets to deal with a seamless handover and assume that a single failed or lost packet is sufficient to trigger a handover.

As far as handover triggering algorithms are concerned, different protocols use different approaches, the simplest metric being evaluating the RSSI values of received packets (SmartHop [6], MobiSense [7], MX-MAC [16]). This approach requires the least complex mechanism to determine link quality fluctuation but it is also unreliable. In [18], the authors propose to combine two metrics, namely, burst loss (consecutive transmission failure) and packet failure rate (pfr). If one of the criteria is fulfilled, a handover procedure is initiated. In [5], the authors propose a handover procedure which employs a fuzzy logic to estimate link quality [1]. It takes packet success rate, link asymmetry, link stability, and signal-to-noise-ratio (SNR) into consideration and combines additional three metrics (energy, traffic load, and depth level) to support a handover. By carefully studying the characteristics of link quality fluctuation in an industrial environment, Zinonos et al. [17] also propose a handover triggering algorithm which employs a fuzzy logic. This approach takes the RSSI values of incoming packets and packet loss rate as its inputs. The output of the algorithm is a trigger decision probability, which, if it falls below a predefined threshold, is used to initiate a handover.

One of the reasons why a handover triggering threshold is required is that a handover entails a seamless neighbour discovery phase wherein a transmitting node searches for an alternative relay node. During this phase, it has to transmit packets in a multicast or broadcast mode, which is inefficient. Almost all proposed handover triggering algorithms rely on an empirically obtained RSSI threshold or the failure of a single packet is sufficient to trigger a handover. An empirical threshold is highly environment dependent. For example, for the CC2420 radio which implements the IEEE 802.15.4 standard, different values are identified: $-75\,\mathrm{dBm}$ in [7], $-87\,\mathrm{dBm}$ in [9] or $-80\,\mathrm{dBm}$ in [13] to achieve a $90\,\%$ packet delivery rate. Thus, it has to be calibrated before deployment, otherwise it may degrade performance. For our evaluation of state-of-the-art in the Evaluation Section, we configure this value to be $-80\,\mathrm{dBm}$ in order to achieve $80\,\%$ of success rate with $95\,\%$ confidence interval (refer to Fig. 2). In this paper, we propose a handover triggering algorithm which does not rely on a predefined threshold. Instead, it equates the cost of packet retransmission (which can be expressed in terms of packet delivery latency or energy) with the cost of a handover and if the former is higher than the latter, a handover is triggered. To compute these costs, our approach seamlessly establishes the statistics of received acknowledgement packets and employs a Kalman filter to characterise and predict the link quality fluctuation.

III. Approach

A quantifiable cost can be associated with every packet transmission a mobile node makes if its communication setting are known at least in a probabilistic sense. This cost can in turn be used to determine the most suitable transmission scheme. If, for example, the transmission should take place in a highly contentious setting, the MAC protocol can elect to turn on the collision avoidance mechanism. If, on the other hand, the medium is less contentious, the collision avoidance mechanism can be turned off because the packet retransmission cost (in case of collision) may be less than the transmission cost of RTS and CTS control packets (which introduce both latency and energy penalty). Similarly, if the cost of a handover is less than the retransmission cost, the mobile node can elect to search for an alternative link and transfer communication to it. The penalty it has to accept is the cost of predicting the link quality and neighbour discovery.

A. Condition for a Handover

Suppose a mobile node has n number of packets to transmit in succession and the expected packet success rate is psr, the retransmission cost, c_{re} (seen only from the mobile node's perspective), can be expressed as:

$$c_{re} = n\left(c_{tx} + c_{rx}\right)\left(1 - psr\right) \tag{1}$$

where c_{tx} and c_{rx} are the transmission and ACK reception cost for a single packet. Similarly, if the node has to communicate with k number of neighbour nodes by sending them m number of packets in order to determine which of them can be the best rely node, the cost it incurs for neighbour discovery, c_s, can be expressed as:

$$c_s = mkc_{rx} \tag{2}$$

where c_{rx} is the cost of receiving a single ACK packet from a neighbour. From Equations 1 and 2, it is clear that a handover is a better option when the quality of a link deteriorates and the packet loss becomes considerably high. In other words, a handover is preferred when:

$$c_{re} > c_s \tag{3}$$

As can be seen, we have expressed the handover condition in a generic sense. The costs may refer to energy, latency, or some other criterion which is important for the application or the user.

B. RSSI and psr

The generic handover triggering condition we specified in Equation 3 implicitly requires the packet success rate. The packet success rate, in turn, is a function of the RSSI values of received packets, but it is impossible to establish a one-to-one relationship between RSSI and psr. Fig. 2 displays the relationship we have established for the CC2420 radio chip after transmitting 450,000 packets in different locations, both indoor and outdoor.

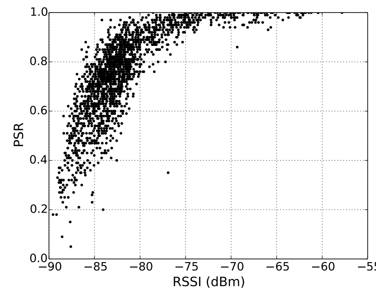


Fig. 2: Packet success rate and RSSI value. psr is calculated for every 100 packets.

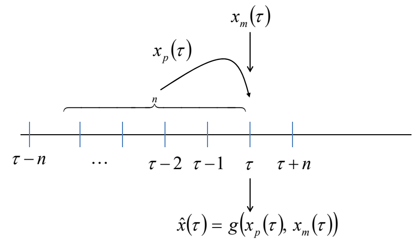


Fig. 3: The steps of estimation to support a seamless handover.

In general, a handover triggering algorithm should deal with three sources of uncertainties: (1) the erratic fluctuation of RSSI values, (2) the uncertainty associated with the relationship between RSSI and psr, and (3) the error associated with predicting the RSSI and psr values of the future, so that a handover can be initiated in a timely fashion.

In order to deal with these uncertainties, we divided packet transmission time into epochs. The average RSSI value of the ACK packets received within an epoch serves as the RSSI value of that epoch. The RSSI and psr values of the past n epochs can be used to determine whether the deterioration of a link quality is a steady phenomenon and therefore a seamless handover should take place in the next epoch.

One of the advantages of dividing time into epochs is that the effect of the three types of uncertainties can be minimised in a systematic way. Specifically, the RSSI and psr values of successive epochs can be regarded as correlated with one another. However, it is impossible to express the RSSI or psr value of epoch τ in terms of past values in a deterministic sense, as they are subject to random fluctuations (we label this error as a process error). Secondly, even if averaging the RSSI values of a single epoch minimises the error associated with the actual RSSI values of the received packets in the epoch, still this estimation contains error, which we regard as a measurement error. The Kalman filter can be employed to combine prediction and measurement values in order to minimise the three sources of uncertainties.

In order to explain our approach, we refer to Fig. 3. Suppose the parameters we wish to estimate at epoch τ can be represented by the generic random variable $x(\tau)$. The reason we describe it as a random variable is that we will never be able to obtain its real value at any given time, owing to the fact that it is subject to the three types of uncertainties. Suppose, at time epoch $\tau - 1$, based on the statistics we have up to that time, we predict the value of x for the epoch τ and label it as $x_p(\tau)$. The index p stands for prediction. At time epoch τ, however, we measure x and label this as $x_m(\tau)$. Both $x_p(\tau)$ and $x_m(\tau)$ contain the actual value of x for that epoch, but each contains a different kind of error. Using the Kalman formalism, we can estimate $x(\tau)$ by properly combining the evidence coming from the two sources:

$$\hat{x}\left(\tau\right) = x_p(\tau) + k(\tau)\left(x_p(\tau) - x_m(\tau)\right) \tag{4}$$

Note that:

$$x_m(\tau) = x(\tau) + v(\tau) \tag{5}$$

where $v(\tau)$ is the measurement error modelled as a random variable. Similarly,

$$x_p\left(\tau\right) = x(\tau) + w(\tau) \tag{6}$$

where $w(\tau)$ is the processor error modelled as a random variable. Hence, our goal should be finding the optimal k such that the difference between the actual $x(\tau)$ and its estimated value, $\hat{x}(\tau)$, is minimum. One way to achieve this goal is minimising the mean square error:

$$e^2\left(\tau\right) = E\left\{\left[x(\tau) - \hat{x}(\tau)\right]^2\right\} \tag{7}$$

The value of k in Equation 4 which minimises the mean square estimation error in Equation 7 is expressed as [2]:

$$k\left(\tau\right) = P_p(\tau)\left[P_p(\tau) + R(\tau)\right]^{-1} \tag{8}$$

where $P_p(\tau)$ is the prediction error covariance, i.e.,

$$E\left\{\left[x(\tau) - x_p(\tau)\right]\left[x(\tau) - x_p(\tau)\right]\right\}$$

which can be expressed as:

$$P_p\left(\tau\right) = P(\tau - 1) + Q(\tau) \tag{9}$$

where $Q(\tau)$ is the process error covariance (to be defined shortly). Finally, $R(\tau)$ is the measurement error covariance, i.e.,

$$E\left\{\left[x(\tau) - x_m(\tau)\right]\left[x(\tau) - x_m(\tau)\right]\right\}$$

respectively, for epoch τ:

IV. Implementation

In order to take both the fluctuation in RSSI values and the psr of received ACK packets, we represent $x(\tau)$ as a vector quantity:

$$x(\tau) = \begin{bmatrix} r(\tau), & psr(\tau) \end{bmatrix}^T \tag{10}$$

where $r(\tau)$ is the RSSI value and $psr(\tau)$ the packet success rate of the epoch τ. Compared to the packet transmission rate, the speed of the mobile node is very small (typically a human movement is below $5\,\mathrm{km/h}$). Hence, for a very short time ($500\,\mathrm{ms}$ to $1\,\mathrm{s}$), the change in the RSSI values of received ACK packets can be approximated as a linear function of time:

$$r(\tau) = a\tau + b \tag{11}$$

from which we have: $r(\tau) = r(\tau{-}1){+}a$. Moreover, compared to the fluctuation in RSSI values, the change in psr between consecutive epochs is imperceptible. Hence, it is plausible to assume that $psr(\tau) = psr(\tau - 1)$. Putting together these two assumption yields:

$$\begin{bmatrix} r(\tau) \\ psr(\tau) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r(\tau - 1) \\ psr(\tau - 1) \end{bmatrix} + \begin{bmatrix} a \\ 0 \end{bmatrix} \tag{12}$$

where a and b are parameters which can be determined by a linear regression [10] and are associated with the covariance between the RSSI values and the time epochs. The two coefficients are determined by minimising the error between the actual and estimated RSSI values using the mean square error estimation. The error associated with our assumption as regards $r(\tau)$ (the linear approximation) and (2) $psr(\tau)$ (the assumption that $psr(\tau) = psr(\tau - 1)$) can be described as the process error, $Q(\tau)$. The process error in $r(\tau)$ can be described by the variance of the past n RSSI values:

$$\sigma^2_{r,p}(\tau) = \frac{1}{n-1} \sum_{k=(\tau-n-1)}^{\tau-1} \left(r(k) - \overline{r}\right)^2 \tag{13}$$

Likewise, the process error as regards the psr can be expressed as:

$$\sigma^2_{psr,p}(\tau) = \frac{1}{n-1} \sum_{k=(\tau-n-1)}^{\tau-1} \left(p_s(k) - \overline{p}_s\right)^2 \tag{14}$$

The process error of the vector $x(\tau)$ expressed as a matrix is:

$$Q(\tau) = \begin{bmatrix} \sigma^2_{r,p}(\tau) & 0 \\ 0 & \sigma^2_{psr,p}(\tau) \end{bmatrix} \tag{15}$$

where we assumed that $\sigma^2_{psr,p}(\tau)$ and $\sigma^2_{r,p}(\tau)$ are uncorrelated. The error associated with the measurement of the actual values of RSSI and the psr for a specific epoch can be determined by taking the variances and covariances of the two random variables for that epoch. Consequently:

$$\sigma^2_{r,m}(\tau) = \frac{1}{m-1} \sum_{i=1}^{m} \left(r^\tau_i - \overline{r}^\tau\right)^2 \tag{16}$$

where $\overline{r}^\tau$ is the mean RSSI value for the time epoch τ. In a single epoch, we have a single psr value, since psr is an average quantity. In order to compute the associated measurement error, we have to take into account the fact that $\sigma^2_{r,m}$ and $\sigma^2_{psr,m}$ are related with one another. This relation

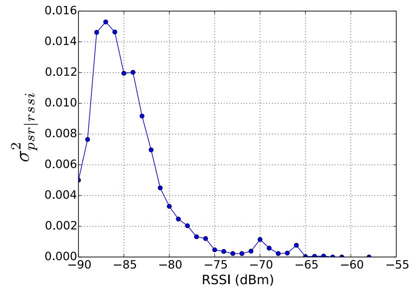


Fig. 4: The conditional variance of psr given RSSI.

is described by the correlation coefficient, from which the measurement error as regards psr can be determined[1]:

$$\sigma_{psr,m}(\tau) = \rho_{r,psr,m}(\tau)\sigma_{r,m}(\tau)\sigma_{psr|r,m}(\tau) \tag{17}$$

where $\sigma_{psr,m}$ corresponds with the error associated with the measured psr for the time slot τ, $\rho_{r,psr,m}$ is the correlation coefficient between the measured RSSI and psr for the time slot τ, and $\sigma_{psr|r,m}$ is the conditional error associated with the psr given RSSI. The quantities in the right term save $\sigma_{r,m}(\tau)$ are determined experimentally, using Fig. 2. Figure 4 displays the conditional psr error as a function of the correlation coefficient and the measurement error associated with the RSSI of epoch τ. Finally, the measurement covariance error is expressed as:

$$R(\tau) = \begin{bmatrix} \sigma^2_{r,m}(\tau) & \sigma_{r,psr,m}(\tau) \\ \sigma_{r,psr,m}(\tau) & \sigma^2_{psr|r,m}(\tau) \end{bmatrix} \tag{18}$$

With $Q(\tau)$ and $R(\tau)$, it is sufficient to compute the Kalman gain for each time epoch and with it, to predict the RSSI and the psr of the future $(\tau+1)$ time epoch. Moreover, with the future values predicted, it is possible to apply Equation 3 and determine whether a mobile node should trigger a handover at epoch τ so that in $\tau+1$ it can switch to a new communication partner.

V. Evaluation

We implemented our handover-triggering algorithm (KMF) and integrated it with the MX-MAC protocol [3]. It runs in a TinyOS runtime environment on the TelosB platform. We also implemented four additional proposed handover-triggering algorithms to make an objective comparison. The first algorithm – Single Packet Failure (SPF) [11] [8] – triggers a handover upon a single packet failure. The Link Loss (LL) algorithm combines consecutive failure and packet failure rate to trigger a handover. Thus, if n packets continuously failed or the packet failure rate falls below a set threshold f within a specified duration, then it triggers a handover. The third algorithm, the RSSI threshold based algorithm (or simply, RSSI), triggers a handover if the average RSSI value of successively received ACK packets drops below a set threshold [6]. The final protocol, MX-MAC [3], takes the RSSI values of present and future epochs into consideration in order to trigger a handover. It implements a normalized LMS filter for predicting the mean RSSI value of a future slot. Table I compares the memory footprint of the different algorithms we implemented.

[1]The strength of correlation between two random variables, $\mathbf{X}$ and $\mathbf{Y}$, can be expressed by the correlation coefficient:

$$\rho_{xy} = \frac{E\left[\mathbf{X} - \eta_x\right]\left[\mathbf{Y} - \eta_y\right]}{\sigma_x \sigma_y}$$

TABLE I: The memory footprint and computational complexity of the different algorithms.

	LL	RSSI	MXMAC	KMF
ROM (bytes)	122	158	5498	4876
RAM (bytes)	8	12	118	98
execution time (ms)	-	-	16 ± 0.03	26 ± 0.05

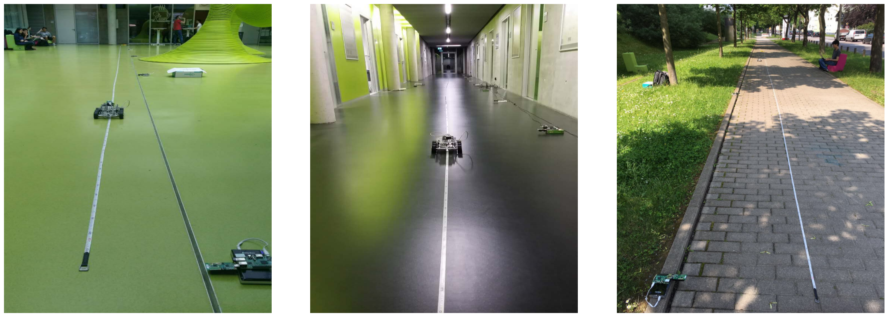
TABLE II: Parameters used for handover trigger experiments in different environment.

parameter	**value**
environment	lobby, corridor, outdoor
spacing between nodes	5 m
speed	0.13 m/s (constant)
motion pattern	straight line walk
duration	160 s
tx-power	-25 dBm, -10 dBm, 0 dBm
IPI	50 ms, 100 ms

In order to evaluate the performance of our algorithm, we conducted a series of experiments using the MobiLab testbed [14]. In our setup, the testbed consisted of 5 static TelosB nodes deployed in a straight line with a $5\,\mathrm{m}$ separating distance between them and a mobile node carried by a robot. We deployed the testbed in a lobby, a corridor, and outdoors (see Fig. 5). The description of our experiment settings is summarised in Tab. II. In order to draw a comparable conclusion for the other handover triggering algorithms, we first launched a large number of preliminary experiments and carried out an in-depth analysis of the received packets. Our aim was to calibrate the parameters for each algorithm (the configuration parameters we obtained are listed in Tab. IV). Afterwards, we executed and repeated each experiment ten times. During each experiment, the robot was moving from one end of the deployment area to the other in a straight line, at a constant speed (approximately $0.13\,\mathrm{m/s}$), whilst the transmitter carried by the robot transmitted packets in burst. As a result, in a single run of experiment, the characteristics of five distinct links could be evaluated (i.e., the communication link established by the robot with each relay node). The runtime characteristics of all sensor nodes and the robot were monitored by using the TFCP framework [15].

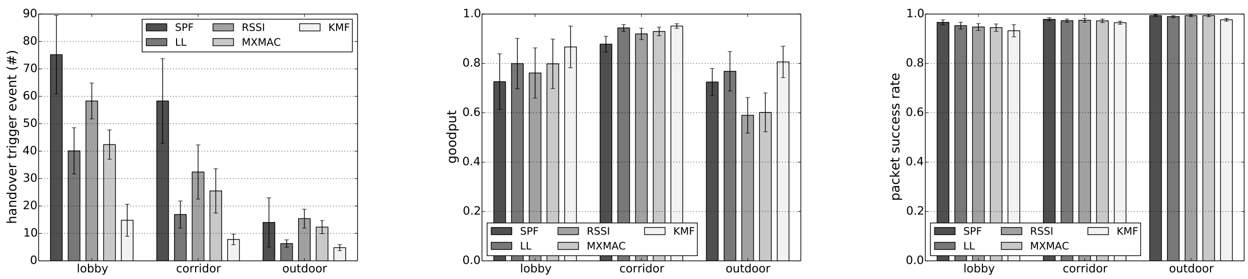
A. Prediction Accuracy

One of the features upon which the performance of our approach depends is the prediction accuracy of the Kalman filter. This feature is important because the implementation



(a) (b) (c)

Fig. 5: The deployment environments for our experiments: (a) Lobby. (b) Corridor. (c) Pathway.



(a) (b) (c)

Fig. 6: Evaluation of the five algorithms in terms of: (a) Number of handover events. (b) Goodput. (c) psr.

TABLE III: Average psr for different links.

environment	link1	link2	link3	link4	link5
lobby	0.67	0.75	0.72	0.70	0.60
corridor	0.87	0.91	0.92	0.95	0.90
outdoor	0.16	0.38	0.42	0.38	0.11

TABLE IV: Config parameters of the different algorithms.

Trigger Algorithm	Trigger Criterion	Slot Size
SPF	single packet failure	-
LL	CF = 2 or PFR > 0.2	10
RSSI	-80 dBm	10
MXMAC	L <= 1	10
KMF	$c_{re} > c_s$	10

of the Kalman Filter added complexity to our algorithm. Tab. III summarises the psr for each link as the ratio of the total number of packets received to the total number of packets transmitted in a link. Fig. 7 summarises the prediction accuracy of the Kalman Filter for the different links, from which it can be seen that the prediction accuracy is above 0.8 (1 being the maximum) for most of the links. Only link 4 and link 5 in the lobby were less than 0.8. This is mainly due to the high false positive (as shown in Fig. 7(c)).

B. Handover Trigger Event

A *handover trigger event* is generated when a handover triggering algorithm initiates a handover as a result of a "belief" by the former that a deterioration in the link quality leads to a disconnection or that the packet loss rate is below a specified threshold. It is a measure of the sensitivity of the triggering algorithm. A highly sensitive algorithm leads to a frequent attempt to transfer a communication to an alternative relay node, and may cause a high handover cost. As most commercially available transceivers are low-powered and low-cost, the RSSI values of received ACK packets may fluctuate for a brief period of time despite the very low speed of the robot. In other words, a fluctuation in the RSSI values of received ACK packets may not necessarily indicate the disconnection of an established link. Thus, the handover triggering algorithm should be tolerant to such transient variations of link quality, otherwise it may lead to a ping-pong handover problem, unnecessarily increasing packet transmission latency and power consumption. Moreover, a mobile transmitter may not be successful in finding a new relay node whenever a handover is initiated, in which case it may waste resources in searching for relay nodes. Fig. 6(a) suggests that our algorithms (KMF) generated a significantly less number of trigger events than all the other algorithms, because it was able to filter transient link fluctuations more efficiently than the other solutions, particularly, in the indoor environments

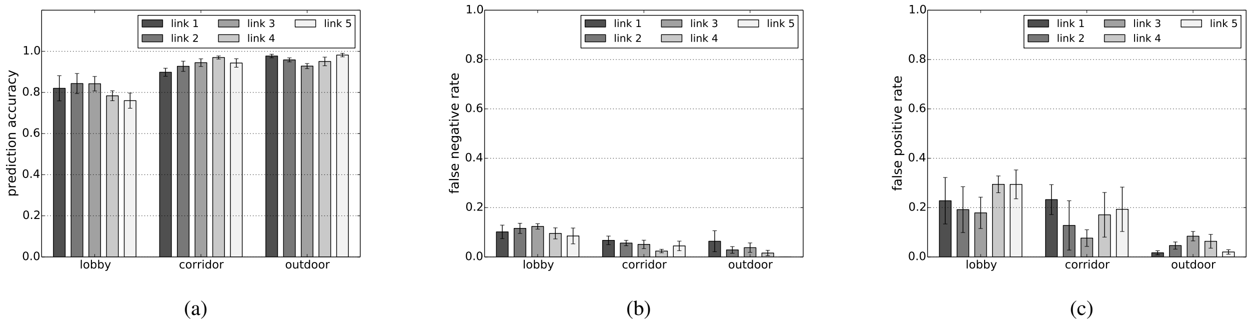


Fig. 7: The prediction accuracy of Kalman filter based handover trigger algorithm applied to 5 different links in three environment: (b) and (c) show false negative and positive respectively

(lobby and corridor). The SPF algorithm performed worst due to its reliance on a single packet failure to trigger a handover.

C. Goodput and Packet Success Rate

We define the *goodput* as the ratio of the number of successfully transmitted data packets to the maximum data packets which can be transmitted in an ideal link during the same transmission period:

$$Goodput = \frac{N_{success}}{N_{ideal}}$$

As shown in Fig. 6(b), KMF gains the highest goodput overall in different environments. The reason is its high data packet transmission efficiency. Furthermore, KMF is the only algorithm the average goodput of which is above $80\,\%$. It can be seen in Fig. 6(c) that, compared to the other algorithms, the performance of KMF degraded a little bit in terms of packet success rate. It achieved $93.2\,\%$, $96.5\,\%$, and $97.7\,\%$ for lobby, corridor, and outdoor, respectively. The reason for the relatively low performance in this respect is its higher tolerance of transient packet failures.

VI. Conclusion

In this paper we proposed a handover-triggering algorithm which takes the RSSI fluctuation and the packet failure rate of a wireless link into consideration. To predict these two quantities, we modelled them as random variables and applied the Kalman filter by dividing time into epochs and analysing the statistics representing their fluctuations in these epochs. Our aim was to tolerate transient fluctuations but accurately foresee middle term trends. We compared our approach with four proposed algorithms, all of which we implemented for the TinyOS and TelosB platforms. As can be seen, compared to all the other schemes, our approach was able to significantly minimise the handover attempt because of its superior prediction technique.

References

[1] N. Baccour, A. Koubâa, H. Youssef, M. B. Jamâa, D. Do Rosario, M. Alves, and L. B. Becker. F-lqe: A fuzzy link quality estimator for wireless sensor networks. In *EWSN*, pages 240–255. Springer, 2010.

[2] W. Dargie. Principles and applications of ubiquitous sensing, 2016.

[3] W. Dargie and J. Wen. A seamless handover for wsn using lms filter. In *LCN*, pages 442–445. IEEE, 2014.

[4] Q. Dong and W. Dargie. Analysis of the cost of handover in a mobile wireless sensor network. In *WWIC*, pages 315–322. Springer, 2012.

[5] H. Fotouhi, M. Alves, A. Koubaa, and N. Baccour. On a reliable handoff procedure for supporting mobility in wireless sensor networks. In *RTN*, 2010.

[6] H. Fotouhi, M. Zúñiga, M. Alves, A. Koubâa, and P. Marrón. Smart-hop: A reliable handoff mechanism for mobile wireless sensor networks. In *EWSN*, pages 131–146. Springer, 2012.

[7] A. Gonga, O. Landsiedel, and M. Johansson. Mobisense: Power-efficient micro-mobility in wireless sensor networks. In *DCOSS*, pages 1–8. IEEE, 2011.

[8] R. Kuntz, J. Montavont, and T. Noël. Improving the medium access in highly mobile wireless sensor networks. *Telecommunication Systems*, 52(4):2437–2458, 2013.

[9] K. Levis. Rssi is under appreciated. In *Proceedings of the Third Workshop on Embedded Networked Sensors, Cambridge, MA, USA*, volume 3031, page 239242, 2006.

[10] J. Neter, M. H. Kutner, C. J. Nachtsheim, and W. Wasserman. *Applied linear statistical models*, volume 4. Irwin Chicago, 1996.

[11] G. Z. Papadopoulos, V. Kotsiou, A. Gallais, P. Chatzimisios, and T. Noël. Wireless medium access control under mobility and bursty traffic assumptions in wsns. *Mobile Networks and Applications*, 20(5):649–660, 2015.

[12] S. S. Rao, B. Kuo, R. W. McCallum, W. D. Chey, J. K. DiBaise, W. L. Hasler, K. L. Koch, J. M. Lackner, C. Miller, R. Saad, et al. Investigation of colonic and whole-gut transit with wireless motility capsule and radiopaque markers in constipation. *Clinical gastroenterology and hepatology*, 7(5):537–544, 2009.

[13] K. Srinivasan, M. A. Kazandjieva, S. Agarwal, and P. Levis. The β-factor: measuring wireless link burstiness. In *SenSys*, pages 29–42. ACM, 2008.

[14] J. Wen, Z. Ansar, and W. Dargie. *MobiLab: A Testbed for Evaluating Mobility Management Protocols in WSN*, pages 49–58. Springer International Publishing, Cham, 2016.

[15] J. Wen, Z. Ansar, and W. Dargie. A system architecture for managing complex experiments in wireless sensor networks. In *ICCCN 2016*, 2016.

[16] T. Zhiyong and W. Dargie. A mobility-aware medium access control protocol for wireless sensor networks. In *IEEE Globecom Workshops*, pages 109–114. IEEE, 2010.

[17] Z. Zinonos, C. Chrysostomou, and V. Vassiliou. Wireless sensor networks mobility management using fuzzy logic. *Ad Hoc Networks*, 16:70–87, 2014.

[18] Z. Zinonos, V. Vassiliou, and C. Chrysostomou. Handoff triggering for wireless sensor networks with performance needs. In *ISCC*, pages 000982–000988. IEEE, 2013.