Journal of Computer and System Sciences 77 (2011) 305–321

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Automatic mobility status estimation in wireless self-organised networks

Richard Anthony *, Mona Ghassemian

Department of Computer Science, The University of Greenwich, Greenwich, London, UK

ARTICLE INFO

Article history:
Received 1 June 2009
Received in revised form 18 September 2009
Available online 1 February 2010

Keywords:
Wireless sensor networks
Self-detection of mobility
Mobility models and metrics
Reinforcement learning
Lévy Flights

ABSTRACT

Wireless sensor networks and ubiquitous computing are rapidly increasing in popularity and diversity. For many applications of these systems the mobility status of devices forms part of the operating context on which self-organisation is based.

This paper describes a novel technique by which wireless devices such as sensor nodes can deduce their own mobility status, based on analysis of patterns in their local neighbourhood. The Self-Detection of Mobility Status algorithm (SDMS) uses a reinforcement learning inspired mechanism to combine the indications from five mobility metrics. For many systems in which a neighbour table is maintained through regular status messages or other interaction, the technique incurs no additional communication overhead. The technique does not require that nodes have additional information such as absolute or relative locations, or neighbourhood node density.

The work considers systems with heterogeneous time-variant mobility models, in which a subset of nodes follows a random walk mobility model, another subset follows a random waypoint mobility model (i.e. has intermittent movement), some nodes have group mobility and there is a static subset.

We simulate these heterogeneous mobility systems and evaluate the performance of SDMS against a number of metrics and in a wide variety of system configurations.

1. Introduction

Wireless sensor networks (WSN) and ubiquitous computing are rapidly increasing in popularity and diversity. Both types of system comprise a mixture of fixed and mobile (or movable) devices. Whilst the two types of system have different core functionalities; the former being primarily concerned with sensing some characteristics of the environment in which it is deployed, and the latter being concerned with using the environment as a means of contextualisation; they both are increasingly concerned with mobile devices. There is also a clear trend in increasing number, variety and capabilities of mobile computing devices and, as a result, there is increasing diversity in the way these devices are used. For many applications of these systems the mobility status of devices forms part of the operating context on which self-organisation is based. Detecting and classifying mobility is complex as there are a wide variety of mobility characteristics, and several models for describing patterns of mobility are popular, although not necessarily sufficiently representative.

Traditionally sensor nodes in WSN are considered to have fixed mobility characteristics for the great majority of implementations. In applications where sensors are physically attached to the infrastructure (e.g. monitoring vibration in a bridge) it is adequate to assume no mobility. The static model may be expanded by introducing mobility in three levels [1] of the sensor network system. It can be the case where the sensors themselves move and collect information, the event monitored

* Corresponding author.
E-mail addresses: R.J.Anthony@gre.ac.uk (R. Anthony), M.Ghassemian@gre.ac.uk (M. Ghassemian).

0022-0000/$ – see front matter
doi:10.1016/j.jcss.2010.01.009

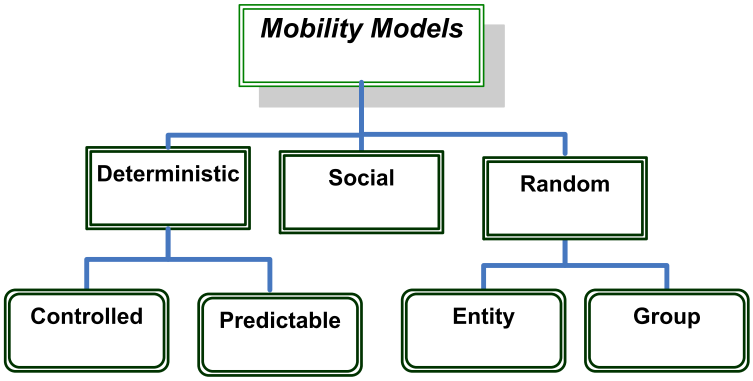

Fig. 1. A summary of mobility models applied in WSN research.

by the network is mobile, or a mobile sink node gathers information from sensors by visiting them and finally relaying the collected data to the sink node.

In some applications such as Zebranet [2] in which sensors are fixed to animals, or other obviously movable objects; mobility is assumed. In cases such as Zebranet, no model of mobility will be perfect, but it is clear that there will be periods of movement punctuated by periods where the node is stationary. Some work assumes mobility status to be implicit in the placement of the sensor (so a sensor attached to a vehicle is mobile and one attached to a building is not, as in [3]) however, vehicles are often static for long periods, and when moving have a very wide range of movement characteristics.

This paper proposes that mobility should be considered a dynamic attribute of sensors and wireless computing devices and that for reasons of scalability, low latency and low communications cost, must be dynamically and locally detected. There are circumstances where assumptions of static and/or global mobility models can be unreliable and this can lead to incorrect or at best inefficient behaviour. In particular, intermittent mobility patterns can be problematic; for example in dynamic clustering, a potentially destabilising situation arises if a node advertises its local cluster and recruits neighbours to it, and then moves off. We suggest that mobility awareness can potentially improve correctness and optimisation in a variety of service and application scenarios.

In order that intermittent mobility can be detected we suggest that it is important to identify mobility patterns on two temporal scales: instantaneous and medium term trend. Group mobility, if detected, offers potential for temporary clustering and cooperation amongst mobile nodes. [4] provides an example (selection of cluster heads) where nodes' class of mobility, if known, can be taken into account.

This work is concerned with automatic, localised, autonomous and low-cost Self-Detection of Mobility Status (SDMS) for nodes in WSN and other related ubiquitous and pervasive applications in which networks self-organise. As more-varied and widespread deployments occur there is a growing need for application behaviour to be contextualised by dynamic mobility status. The mobility information calculated by the proposed algorithm can be used for several purposes, examples include: optimising the performance of protocols sensitive to dynamic environments (such as routing); and assisting vendor-specific services such as service discovery and customisation for individuals or groups of mobile users.

This paper is organised as follows: Section 2 discusses background and related work, Section 3 describes the SDMS algorithm, Section 4 deals with the simulation model and evaluates the algorithm's performance. We then conclude in Section 5 and identify areas for further investigation in Section 6.

2. Background and related work

2.1. Mobility models

Several mobility models are commonly used to describe patterns of mobility in WSN. In this paper we use these popular models, briefly discussed below, to analyse the performance of our proposed algorithm.

Mobility models can be divided into three primary categories: random [5], deterministic and social [6] mobility models. A summary of the mobility models is presented in Fig. 1.

Random mobility models define arbitrary movement without constraints and can be further subdivided into entity and group mobility models. In an entity model nodes are moving within a defined area with a randomly chosen speed, direction and pause time. Some of the entity mobility models are random waypoint, random walk, random direction. Variants include city section, Voronoi, freeway and Manhattan models. Group mobility, described by for example the RPGM model (see below), encompasses a mobile reference point in which mobile nodes belonging to the same group exhibit similar mobility characteristics such as speed and direction (e.g., group searching for a target scenario). Random models, unlike social and deterministic mobility models, assume open, unobstructed areas in which nodes are free to move according to the constraints of the mobility model. In real-world scenarios, it is rare that groups of people are located in completely unobstructed areas; there are typically buildings, vegetation, benches, cars, and other objects that obstruct one's path. Additionally, it is unlikely to be the case that people follow random trajectories. On campuses people tend to follow provided pathways, in cities people follow sidewalks, in buildings people are confined to hallways, etc. While occasionally individuals may stray from the provided pathways, the majority of movement typically occurs along these paths. The models used in our analysis are defined in more detail in the following:

- *Random Waypoint Model (RWP)*: This model includes pause times between changes in direction and/or speed [6]. A mobile node (MN) begins by staying in one location for a certain period of time (i.e., a pause time). Once this time expires, the MN chooses a random destination in the simulation area and a speed that is uniformly distributed between [*minspeed, maxspeed*]. The MN then travels toward the newly chosen destination at the selected speed. Upon arrival, the MN pauses for a specified time period before starting the process again. The RWP model is not universally popular as some believe it to be unrealistic [7], but it has been widely used in many analyses in MANETs research.
- *Random Walk Model (RWK)*: This model [6] is a simplified version of the Random Waypoint mobility model where pause time between the random motions is set to zero.
- *Lévy Flight Mobility Model* is a type of random walk in which the increments are distributed according to a heavy-tailed probability distribution. After a large number of steps, the distance from the origin of the random walk tends to a stable distribution.
- *Reference Point Group Model (RPGM)*: This model represents the random motion of a group of MNs as well as the random motion of each individual MN within the group [6]. Group movements are based upon the path travelled by a logical centre for the group. The logical centre for the group is used to calculate group motion via a group motion vector. The motion of the group centre completely characterises the movement of its corresponding group of MNs, including their direction and speed. Individual MNs randomly move about their own pre-defined reference points, whose movements depend on the group movement. As the individual reference points move from time t to $t+1$, their locations are updated according to the group's logical centre. Once the updated reference points at $t+1$ are calculated, they are combined with a random motion vector, to represent the random motion of each MN about its individual reference point.

The social mobility models are based on relationships between people. This class of model defines the movement between different social groups. It is not difficult to see by eye that the random mobility models generate behaviour that is most unhumanlike. To use the social relationships among individuals, groups of hosts need to be defined that move together. Sociability factors indicate the attitude of an individual towards interaction with others. A host that belongs to a cloud moves inside it towards a goal (i.e., a point randomly chosen in the cloud space) using the standard random waypoint model. When a host reaches a goal, it also implicitly reaches a decision point about whether to remain within the cloud and, if leaving, to where it should go. Mobility models founded on the relationships between people have been presented in [5] and [6].

- A *community based mobility model* is based on the social network theory [8]. One of the inputs of this model is the social network that links the individuals carrying the mobile devices; this is used to generate realistic synthetic network structures. This model allows collections of hosts to be grouped together in a way that is based on social relationships among the individuals. This grouping is only then mapped to topographical space, with topography biased by the strength of social ties. The movements of the hosts are also driven by social relationships among them.
- Recent work into human mobility patterns has identified Lévy Flights [9] as a possible representation of mobility. The distribution follows a power law and is long-tailed such that the movement tends to consist of many small localised steps, punctuated by fewer longer steps to a new vicinity. However the current studies such as [10] have focussed on mobile phone usage data. Here the positional granularity is that of the mobile phone network cell spacing, and the temporal granularity is dependent on call event occurrence and thus the data is an incomplete record of movement. The findings seem to indicate that the Lévy Flight partially describes the movement patterns, although the larger displacements are statistically fewer than expected. This interesting work raises issues for mobility within the smaller distance scales of typical WSNs. In particular we envisage Lévy Flights (Fig. 2A) could help represent movement in WSNs at two levels: (1) the trajectory is within the WSN boundary (Fig. 2B); (2) the WSN is a vicinity through which the trajectory passes (Fig. 2C). We take these possibilities into account in our simulations and the analysis of results in Section 4.

The deterministic mobility models predefine the ability of a node to move intentionally and can be divided into two classes: predictable; and controlled (or robotic) mobility models. In predictable mobility patterns the mobility path of the mobile nodes is known deterministically according to the application scenario and is actively controlled in real time. Predictable mobility enables a whole new set of possibilities in sensor networks where nodes can be deployed at optimal locations for monitoring in industrial application scenarios. Another 'predictable mobility' scenario is public transportation vehicles (buses, shuttles and trains), which can act as mobile data collectors in WSNs. In controlled models, the path of the mobile node is defined to optimise the protocol performance. An application scenario is the motion of data collector nodes (intelligent mobile nodes) in the WSN to disseminate data. This type of node acts as an agent for collecting and routing data generated on the sensor nodes. Sensors are assigned weights based on the amount of data they generate and the latency of

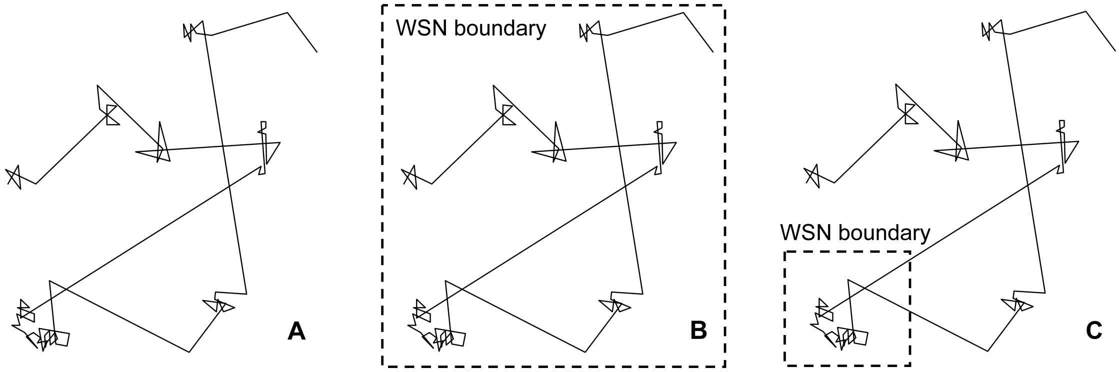

Fig. 2. Lévy Flights and their possible relationship to mobility in WSNs. A: A flight at the scale of a city, in line with the work in [10]. B: A possible flight at the level of a single WSN, for example visiting shops within a shopping mall. C: A possible hybrid case where a mobile device is carried on a flight that passes through a WSN that can detect it.

the data already present on the sensor. The weights assigned to the sensors are used as a basis for deciding the motion of the data collector.

In the presence of mobility an appropriate selection of relevant mobility models should be taken in to account for the protocol performance evaluations. This is to ensure that the various characteristics of mobility present in the studied system are represented in the evaluation. We have used a number of the discussed mobility models to create different scenarios to measure the accuracy of the algorithm proposed in this paper.

2.2. Mobility status as dynamic context

This section discusses application scenarios in which mobility information adds value, and related work that has applied sensor nodes' local mobility status to improve performance.

Sensor nodes' continuously locally determined mobility status could facilitate improved effectiveness of WSN applications in various ways. Examples include:

1. Identifying potential data mules to carry data between isolated groups of devices [11], as in 'pocket switching' with mobile phones [12]. In [13] sensor nodes on vehicles are used to convey data between networks in a town environment where mobility status is implied by object type, i.e. vehicle implies mobile.
2. A node's mobility status may serve as context information to a WSN application, e.g. logging the pattern of movement of the equipment that the sensor is attached to. This relates to applications in which the sensor location contextualises sensed data, see for example [14]. In location-sensitive applications, dynamic mobility knowledge may be necessary for correct operation.
3. WSN routing uses knowledge of nodes' locations or relative positions to efficiently route messages, e.g. [15,16]. Route reliability could be improved by avoiding routing via moving or likely-to-move nodes.
4. Dynamic mobility information can enhance cluster management; e.g. [17] assumes static topologies for simplicity when selecting the 'centre' node of a cluster as the cluster head – undetected mobility could be destabilising.
5. Dynamic sensor mobility status can potentially complement the use of relative sensor position data in object tracking applications, such as described in [18].
6. The mobility status of nodes can be integrated to give a neighbourhood-level or possibly global measure of mobility. There may be opportunities to perform high-level or application tuning based on the extent of detected mobility at the system level. Examples include de-tuning in more volatile conditions, or increasing beacon messaging when there is a higher degree of mobility, as nodes are in contact for shorter periods of time. Mobility detection based on early work towards SDMS was explored in the Emergent Cluster Overlay protocol (ECO) [19]. ECO self-configures WSN nodes into clusters of a dynamic application-specified size, and automatically allocates cluster heads. Dynamically detected mobility state information is used to prevent a mobile node being selected as a cluster head and thus prevents instability arising from ephemeral clusters being advertised (as the mobile node passes by). When a node detects either itself, or a neighbour is mobile the status message beaconing rate is increased.

As discussed above, previous work includes some systems that do differentiate between mobile and static nodes, but this is generally either based on pre-assigned device types (such as fitted to vehicles vs roadside equipment as in [3]) or based on more-powerful nodes equipped with GPS and sustainable (solar) power supplies as in [2], where mobility is implicit in the application. However, typical sensor nodes are low power, low cost devices designed to be deployed in large numbers, as such, these devices typically do not have in-built GPS and may even be deployed in situations (such as underground tunnels, within dense buildings or under sea; attached to oil-platforms, small underwater vessels, etc.) where GPS reception is not available. It is also attractive to be able to deploy nodes without having to explicitly configure individuals based on their placement.

This work proposes a novel mobility detection method which is suitable for deployment on the simplest of sensor platforms as it deduces mobility status based only on information contained in local-neighbour tables built up through neighbour discovery mechanisms and receipt of status or heartbeat messages.

The SDMS algorithm is based on patterns (of neighbour discovery, disconnection, mean association longevity, etc.) in the makeup of a node's local neighbourhood. Our approach is clearly differentiated from techniques based solely on the statistical information of the physical link durations. For example, analysis of random waypoint based on the physical link duration and stability is discussed in [20]. Relying on the physical link duration statistics can hide the details of the mobility model. The SDMS approach is general enough to cover different mobility models.

In addition to identifying individual mobility status, SDMS is able to detect situations where the node is part of a small group travelling together – for example co-located on a moving trolley or vehicle. Instantaneous mobility status and mobility trend information are made available to higher level self-organisation protocols which can use the information to improve their correctness, efficiency or optimality.

3. The SDMS algorithm

Mobility detection mechanisms estimate the level and the direction of mobility in wireless self-organised networks. Three categories of mobility estimation method are identified in the literature:

- Received signal strength indication, derived from physical layer measurements.
- Predicting from historical values based on underlying localisation mechanisms.
- Calculating from messages received from neighbours.

The first mechanism can design a lightweight mobility estimation mechanism. However signal issues such as reflection, multipath and shadowing result in inaccurate distance measurements. The second method relies on a localisation mechanism which provides more accurate estimation at the expense of high computational cost. This method is not suitable for low computational resource sensor nodes, and also requires an accurate time reference to measure time of arrival (ToA) of the packets. A two-way ranging method has relaxed this requirement by doubling the cost of packets to measure ToA between two nodes.

The proposed SDMS algorithm fits into the third category which requires nodes to announce their presence at regular intervals. Mobility estimation accuracy improves as the interval between updates is reduced [21], however this is a mechanistic tuning aspect, outside of the algorithm itself. The level of required accuracy depends on the sensitivity of protocols that take this information as input for their operation.

SDMS is designed to operate in systems containing a mix of static and mobile nodes, in which nodes are not explicitly aware of their mobility and mobility is not implicit by the type or ID of the sensor. Low-budget, small footprint and low-power WSN nodes do not have location or mobility detection sensors. In addition to detecting relative mobility between pairs of nodes (this could be described as first order mobility detectable from interaction), SDMS combines several mobility metrics to detect absolute mobility (i.e. this could be described as second order mobility, as detected from interaction). This operates on a completely distributed and autonomic basis, with each node determining its own mobility state and requiring no a priori mobility information.

A node may be sometimes mobile and sometimes stationary; for example it may be attached to an object that is moved around inside an environment (hospital, airport, factory, shopping mall, etc.) but is also left in a fixed position for lengthy periods; thus the mobility attribute must be dynamically determined. Nodes may travel in groups either permanently or temporarily. Examples include: several items with attached sensors may be placed on the same trolley or conveyor belt; vehicles with sensors may be caught up in slow moving traffic; sensors on a tractor and trailer which can be linked together; and sensors carried by people or animals that may sometimes flock together.

3.1. Software architecture

SDMS is designed to support protocols that are used to build a stack of functionality, with primitive services and instrumentation at the low levels and applications which use the provided information at higher levels. In this way several protocols can share the same information and more-importantly, the communication costs. The algorithm provides mobility status information to higher level self-organisation protocols via a simple API or via shared variables if integrated at the code level (as is intended, for efficiency). Fig. 3 shows the placement of SDMS into a sensor node software stack. The shaded block represents the SDMS interface by which the generated mobility state indicators are made available to any of the protocols in the higher layers.

Knowledge of mobility status is based on characteristics of the node's neighbourhood including for example the ratio of long-term to short-term neighbours, and the rate of new-neighbour discovery. These mobility metrics are derived from the neighbour table with very low computational cost as explained in Sections 3.4, 3.5, and 3.6.

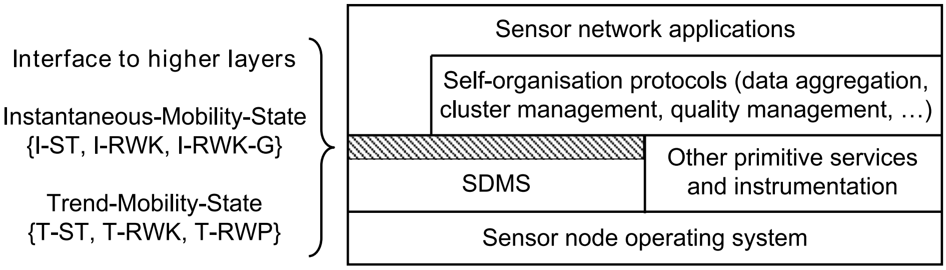

Fig. 3. SDMS as part of a sensor node software stack.

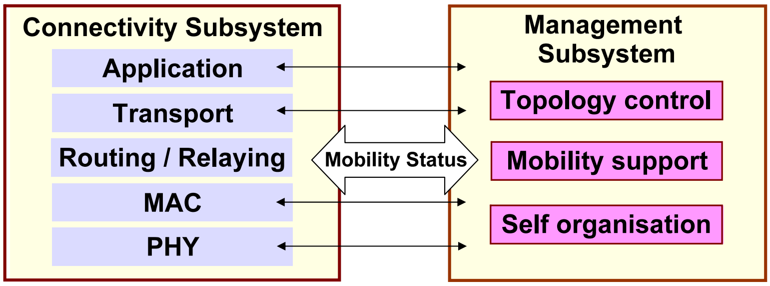

Fig. 4. Logical architecture of the protocol stack.

The algorithm requires that some form of single hop heartbeat/beacon/status messages are regularly broadcasted in order that neighbour tables are kept up to date. Examples of WSN self-organisation protocols where this 'status message' functionality is explicitly provided include the Emergent Graph Colouring algorithm [22] and the Emergent Cluster Overlay protocol [19]. For SDMS purposes, the content of such a message only needs to include a unique node ID. If such status messaging functionality is already present in the system, SDMS introduces no further communication requirement. However it is necessary that the maintenance of the name table includes adding time-of-receipt timestamps and recording the time that each neighbour is originally discovered; these values need to be stored in the neighbour table. Note that a sleeping or dormant node can only be detected by SDMS if it continues to generate status messages.

The mobility estimation information derived from the status message information in the network layer can be used to adapt the behaviour (e.g. transmission rate) of the application layer protocols. Therefore the proposed algorithm can be described as a cross-layer solution as illustrated in Fig. 4.

3.2. Mobility model parameters

Random mobility models define arbitrary movement without constraints and can be further subdivided into individual and group mobility models [6]. In an entity model nodes are moving within a defined area with a randomly chosen speed, direction and pause time. We chose random waypoint and random walk (each with random direction) models for our analysis. Group mobility models encompass a mobile reference point in which mobile nodes belonging to the same group exhibit similar mobility characteristics such as speed and direction (e.g. group searching for a target scenario). To maximise the utility of the mobility status information to higher layer protocols, SDMS provides two types of mobility parameters. *Instantaneous* and *Trend* models operate at each node; patterns in the instantaneous mobility state influence the trend mobility state.

Instantaneous mobility Three mobility states ('I-states') are defined {I-ST, I-RWK, I-RWK-G} with the following meanings: I-ST (the node is STationary at this moment); I-RWK (Random-WalK, the individual node is moving independently of other nodes at this moment); and I-RWK-G (the node is moving as part of a Group with at least one other node at this moment). By continuously integrating recent instantaneous mobility states over time the node's mobility trend is determined.

Trend mobility Three mobility states ('T-states') are defined {T-ST, T-RWK, T-RWP} with the following meanings: T-ST (the node has been predominantly stationary recently at the instantaneous level); T-RWK (the node has been predominantly mobile recently at the instantaneous level); and T-RWP (Random WayPoint, the node has recently had long or many periods of movement punctuated by long or many periods of being stationary).

The *Actual mobility* status of a node is used in the model to control the node's movement, and to determine the prediction accuracy of the algorithm. Five actual-mobility categories are defined {A-ST, A-RWK, A-RWK-G, A-RWP, A-RWP-G} with the following meanings: A-ST (the node is always STationary); A-RWK (the individual node follows a Random-WalK independently); A-RWK-G (the individual node follows a Random-WalK as part of a group); A-RWP (the node has periods of movement punctuated by periods of being stationary, and moves independently); A-RWP-G (the node has periods of movement punctuated by periods of being stationary, and moves as part of a group).

3.3. Mobility metrics

To find a node-local, mobility model independent and efficient solution, we investigated a wide variety of neighbourhood-related local metrics of mobility status. These are derived or inferred entirely from information in the neighbour table (including the occurrence of updates, and timestamps of message receipt). The considered metrics include: neighbour degree (number of neighbours), the rate of new-neighbour discovery, the rate of existing neighbour disconnection, the length of intervals when no neighbour table updates occur (indicating temporary isolation), the number of long-term neighbours, the ratio of new to long-term neighbours, and the mean time neighbours remain in contact.

The various metrics were compared in terms of their ability to distinguish between stationary and mobile nodes, nodes moving in groups, and also punctuated mobility where nodes or groups of nodes have intermittent movement. The metrics were evaluated over a wide environment state space by varying: the mean speed of mobile node movement; the mean randomness of trajectories (with trajectory change intervals of between 3 and 300 seconds); the mean neighbour degree; the proportion of nodes that have each of the various classes of mobility; and for intermittently-mobile nodes, the ratio between periods moving and pause-time.

The simulation investigation revealed that each mobility metric has different strengths in different circumstances and also that no single metric could reliably distinguish between all the different mobility classes under all circumstances. To avoid over-complex tuning issues, metrics that offered only marginal benefits were discarded. The final configuration is stable over a wide range of conditions and uses five mobility metrics in combination:

1. Occurrence of isolation periods. Isolation is defined as a period in which no neighbour table updates occur, i.e. no messages are received. This can help to detect isolated static nodes, and is a strong negative indicator for group membership.
2. The number of long-term neighbours (i.e. number of neighbours that are continuously known to the node since its start up time). The existence of such is an excellent negative indicator for independently mobile nodes with either continuous or intermittent mobility.
3. The fraction of total neighbours that are long-term known. This is found to be a very good discriminator between I-ST and I-RWK-G nodes in topologies with certain density characteristics (where the former tend to have a majority of long-term neighbours and the latter tend to have a smaller fraction of long-term neighbours and a higher rate of new-neighbour discover and disconnect events).
4. Neighbour longevity index (the mean time all current neighbours have been continuously known). This was found to be one of the best metrics when used alone. This is a good general differentiator between all three I-states and is particularly good at detecting I-RWK nodes at low movement speed. The interpretation of the values is fuzzy and requires careful tuning: high values indicate I-ST nodes and low values indicate I-RWK otherwise I-RWK-G is indicated.
5. Neighbourhood changes (link changes, i.e. neighbour discovery and disconnection). This metric combines the number of new neighbours discovered recently and the exponentially smoothed disconnect rate, where nodes move out of communication range and lose contact. This metric can identify temporary isolation and contributes to the classification of static nodes in sparse parts of a topology which only have neighbours when a mobile node or group passes by.

3.4. Operation

SDMS operates autonomously on each sensor node. Nodes are not initially aware of their physical or logical locations with respect to their neighbourhood. When a node receives a beacon/status message it either discovers a new neighbour or gets an update of the state of a previously known neighbour. The algorithm operates continuously, i.e. even if a steady state is reached it is not recognised as such, and there is no termination point; due to the dynamic nature of the system and the fundamental notion that we consider mobility to be time variant. Neighbour details are removed from the neighbour table if they are not heard from for a period greater than two times the normal beacon (status) message period T_{Beacon}.

Fig. 5 provides a block-level overview of the algorithm.

SDMS operates in three stages.

Stage 1 is metric generation (box B in Fig. 5), based on information held in the neighbour table (box A in Fig. 5). For some metrics this involves exponential smoothing of the series of instantaneous values (this aspect is discussed later).

Stage 2 is the determination of instantaneous mobility state (I-state). To achieve this, the algorithm uses a reinforcement learning inspired mechanism to combine the indications from the five mobility metrics.

On a periodic basis an instantaneous Confidence Value (CV) is determined for each I-state. These CVs represent the level of belief that the node is currently in the respective mobility state.

Each of the metrics described in Section 3.3 contributes to the confidence of being in one or more of the I-states. This is because the individual metrics are not perfect discriminators under all conditions, so with regard to the metric values the

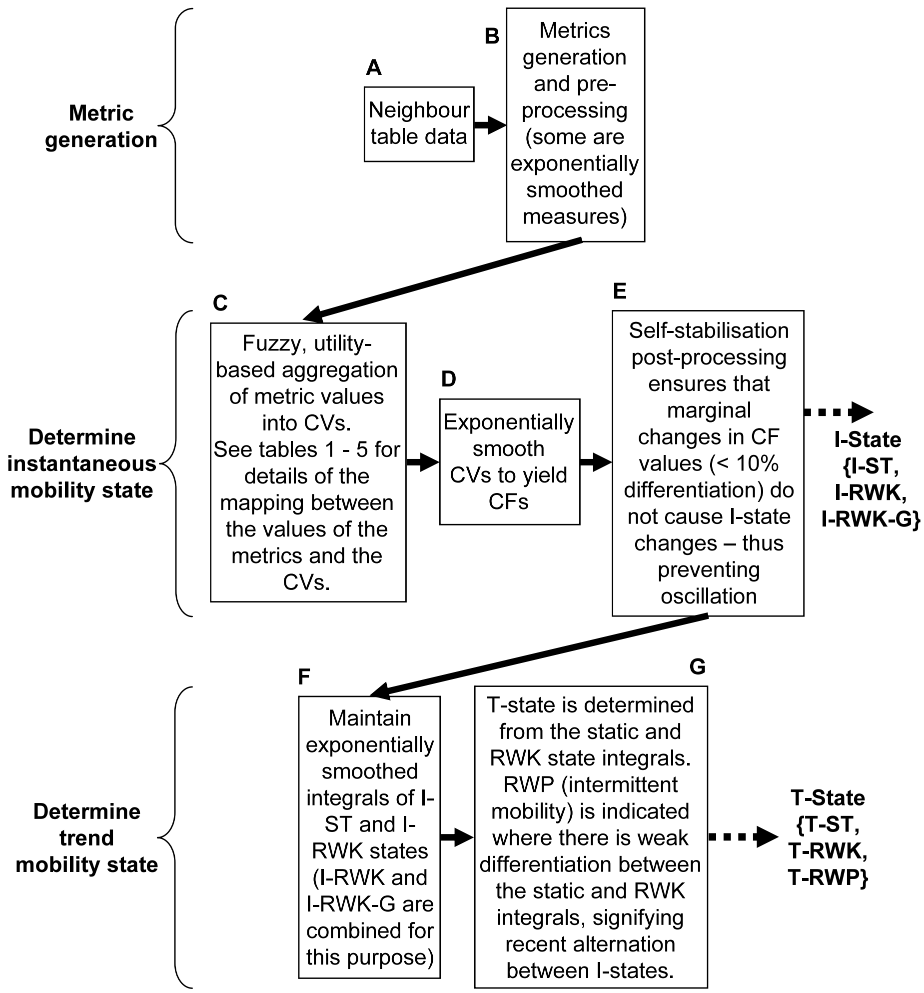

Fig. 5. Block-level overview of the SDMS algorithm.

states can overlap. This is why the five metrics are needed; each has particular utility within a subset of the discrimination state-space and collectively they cover all the possibilities.

The combination of the mobility metrics uses a fuzzy logic technique (box C in Fig. 5). The value of each metric either increases, decreases or leaves unchanged the CVs (which are initialised to 0 at the start of each evaluation step), based on pre-defined weight-mapping matrices which are shown in Tables 1–5.

A Confidence Factor (CF) is maintained for each I-state by exponentially smoothing the sequence of CV values (box D in Fig. 5). This decouples the causal events from direct state transitions and enables recent patterns in behaviour to influence

Table 1
CV update matrix for the ‘isolation’ metric.

Isolation (indicated by a threshold period since last message received)	Effect on instantaneous confidence values (CV)		
	I-ST_{CV}	I-RWK-G_{CV}	I-RWK_{CV}
True	+0.2	−0.5	+0.1
False	no effect	no effect	no effect

Table 2
CV update matrix for ‘number of long-term neighbours’ metric.

Number of ‘long-term’ neighbours	Effect on instantaneous confidence values (CV)		
	I-ST_{CV}	I-RWK-G_{CV}	I-RWK_{CV}
0	no effect	no effect	+0.2
⩾1	+0.1	+0.1	no effect

Table 3
CV update matrix for 'proportion of long-term neighbours' metric.

Proportion of neighbours that are 'long-term'	Effect on instantaneous confidence values (CV)		
	$I\text{-}ST_{CV}$	$I\text{-}RWK\text{-}G_{CV}$	$I\text{-}RWK_{CV}$
$\geqslant 0.00$, $\leqslant 0.30$	no effect	+0.1	no effect
>0.30, <0.50	+0.1	+0.1	no effect
$\geqslant 0.50$, $\leqslant 1.00$	+0.1	no effect	no effect

Table 4
CV update matrix for 'neighbour longevity' metric.

Neighbour longevity metric value	Effect on instantaneous confidence values (CV)		
	$I\text{-}ST_{CV}$	$I\text{-}RWK\text{-}G_{CV}$	$I\text{-}RWK_{CV}$
>0.70, $\leqslant 1.00$	+0.2	no effect	no effect
>0.40, $\leqslant 0.70$	+0.1	+0.2	no effect
>0.25, $\leqslant 0.40$	no effect	+0.1	+0.2
$\geqslant 0.00$, $\leqslant 0.25$	no effect	no effect	+0.2

Table 5
CV update matrix for 'neighbourhood changes' metric.

Recent neighbourhood changes	Effect on instantaneous confidence values (CV)		
	$I\text{-}ST_{CV}$	$I\text{-}RWK\text{-}G_{CV}$	$I\text{-}RWK_{CV}$
Number of new neighbours = 0 AND neighbour disconnect rate < 0.1	+0.5	no effect	no effect
otherwise	no effect	no effect	no effect

the outcome. A simple self-stabilisation technique is used to reduce oscillation caused by flipping between detected mobility states in marginal cases (box E in Fig. 5). An I-state change only occurs when one CF has a differential of at least 10% over both of the other I-state CFs (1).

$$\begin{aligned} \text{I-state}_t = {} & \text{I-ST} && \text{where} \quad CF_{\text{I-ST}} > CF_{\text{I-RWK}} * 1.1 && \text{and} \quad CF_{\text{I-ST}} > CF_{\text{I-RWK-G}} * 1.1 \\ & \text{I-RWK} && \text{where} \quad CF_{\text{I-RWK}} > CF_{\text{I-ST}} * 1.1 && \text{and} \quad CF_{\text{I-RWK}} > CF_{\text{I-RWK-G}} * 1.1 \\ & \text{I-RWK-G} && \text{where} \quad CF_{\text{I-RWK-G}} > CF_{\text{I-ST}} * 1.1 && \text{and} \quad CF_{\text{I-RWK-G}} > CF_{\text{I-RWK}} * 1.1 \\ & \text{I-state}_{t-1} && \text{otherwise} \end{aligned} \tag{1}$$

Stage 3 is the determination of the longer-term mobility trend indication (T-state). This is derived from the aggregation of instantaneous I-state values over time (box F in Fig. 5), yielding trend mobility indicators $\text{T-ST}_{\text{TREND}}$ and $\text{T-RWK}_{\text{TREND}}$. The longer-term mobility trend prediction is not concerned with group mobility, since this attribute can be determined from the current I-state value. Instead, at the longer term trend level it is more interesting for applications to be able to differentiate between stationary, mobile, and punctuated mobility, as these three states map onto popular identifiable mobility models. Since the I-RWK and I-RWK-G states both indicate instantaneous mobility they are combined for the purpose of trend mobility determination.

Where there is strong differentiation between the $\text{T-ST}_{\text{TREND}}$ and $\text{T-RWK}_{\text{TREND}}$ values then the higher value determines the mobility trend, T-ST or T-RWK respectively. However, weak differentiation implies that there have been some periods of movement and some static periods (i.e. punctuated mobility) thus the T-RWP trend is indicated (box G in Fig. 5). This is a fuzzy aspect which required careful tuning for differentiation optimised for a wide range of conditions, see (2).

$$\begin{aligned} \text{T-state} = {} & \text{T-ST} && \text{where} \quad \text{T-ST}_{\text{TREND}} \geqslant 2\ \text{T-RWK}_{\text{TREND}} \\ & \text{T-RWK} && \text{where} \quad \text{T-RWK}_{\text{TREND}} \geqslant 4\ \text{T-ST}_{\text{TREND}} \\ & \text{T-RWP} && \text{otherwise} \end{aligned} \tag{2}$$

3.5. Tuning parameters

The SDMS algorithm has several tuning parameters which allow the approach to be customised for specific environments, such as where it is known that any mobile nodes will move at a certain speed, or what the proportion of nodes that are mobile will be. The various weights by which the metrics update the CVs (see Tables 1–5) can be adjusted to favour particular metrics. However, the values presented in this paper have been carefully determined so as to provide accurate predictions over a wide space of mobility scenarios (as explored in the evaluation section later) and thus ensure general applicability.

Table 6
Configuration of the exponential smoothing.

Metric	Smoothing factor α	Responsiveness to turning points, in terms of lag, which approximates to $1/\alpha$ sample periods
SDMS Stage 1: Neighbourhood change ('Neighbour disconnect' subcomponent of metric)	0.25	4
SDMS Stage 2: CV values smoothing to determine CF values	0.125	8
SDMS Stage 3: Trend prediction	0.025	40

An exponential smoothing technique has been used within each of the three stages of the algorithm's operation. This particular technique has been used to achieve a moving average in which the most recent value has constant weight and older values have correspondingly decreasing weight. This is important because the algorithm is designed to operate continuously, and with the type of prediction desired, the more-recent behaviour has higher value in the prediction process.

The smoothed value of the metric approximates to

$$m = \alpha X_t + (1-\alpha)X_{t-1} + (1-\alpha)^2 X_{t-2} + (1-\alpha)^3 X_{t-3} + \cdots \tag{3}$$

where m is the moving average, α is the smoothing factor ($0 < \alpha < 1$), X is the value of the metric concerned and t is the sample time.

Selection of the smoothing factors provides another tuning opportunity. However, care must be taken when choosing values of α; higher values make the function more sensitive to individual events, but can lead to unstable predictions. On the other hand lower values provide better smoothing as they 'increase the memory' of the function, but at the same time increase lag (which approximates to $1/\alpha$ sample periods). The smoothing factors used (see Table 6) were selected carefully to provide a suitable compromise between sensitivity to events and smoothing effect appropriate at each level of the algorithm's operation so that the overall behaviour is optimised for a wide range of system conditions. The selection of these values was based on analysis of simulation results at the algorithm development stage. In each case, the tuning process focussed on finding the largest α that gave satisfactory results, because the predictions need to be as timely as possible for use in WSNs. For example, the trend prediction uses a smaller value of α so that the trend prediction takes into account a longer recent history (and gives a more stable result, as the effects of sudden changes are throttled), whereas it was found that a much larger smoothing factor could be used with the CF values (making the I-state predictions more responsive) without any significant degradation in accuracy.

3.6. Resource efficiency

The use of fuzzy aggregation of metrics and an exponential smoothing technique ensures that the algorithm is very efficient in terms of resource usage; requiring the storage of one floating point number for each CV and CF value. The exponential smoothing lends itself to a lightweight codification in which it is necessary to retain only the most recent moving average value (m) and not the history of recent values as with other moving average techniques; saving memory. The function (3) is codified as an iterative method (4).

$$m_{t+1} = \alpha X_{t+1} + (1-\alpha)m_t \tag{4}$$

Processing overheads and latency are improved by not having to work through a table of historic values. A small amount of additional processing and storage is associated with the generation of the five mobility detection metrics. The data storage overhead for the algorithm is of the order of 150 bytes (assuming a floating point number requires 4 bytes), including weight values but not including the neighbour table. These considerations are very important for WSN nodes where resource efficiency is a major concern.

In [23] a hybrid network comprising an IEEE 802.11 wireless LAN and an ad hoc routing scheme is proposed in which topology construction is mainly achieved using periodic beacons. As an observation, the overhead of the topology discovery process is reported to be a linear function of the beacon rate. SDMS is based on such beacons for estimating the mobility information required for adaptation of the behaviour (e.g. transmission rate) of the application layer protocols. Hence, in systems where such beaconing is already present in lower layer protocols, no extra overhead is generated to the network for the proposed mechanisms. Where the beacon must be provided specifically for SDMS the overhead is one minimum sized packet per node per second.

4. Analysis

This section describes the simulation analysis of our proposed algorithm. We implemented a simulator [24] as the first step to collect independent measurement to study the effect and accuracy of the proposed algorithm.

4.1. The simulation model

The current simulation model assumes that all nodes have the same radio communication range. This is realistic for typical current WSN deployments, but is not a limitation of the algorithm (exploration of heterogeneous radio transmission power is left for further work). Randomly generated topologies containing up to 1000 nodes are supported.

Protocol-level events such as message receipt occur on a timescale of the order of 1000 ms (i.e. the value of T_{Beacon} for the simulation experiments). To ensure highly accurate simulation, including effects such as short message delay, the model operates in fine-grained discrete steps each representing 1 ms of elapsed time.

SDMS has been designed to cope with various degrees and types of mobility. To analyse the performance of the proposed algorithm we have applied a selection of the mobility models described in Section 2.1. The model has been designed to be highly flexible, supporting a very wide exploration of the algorithm's performance in a wide variety of system configurations. The model can be user configured via the user interface (in GUI mode) and for repeatable and detailed investigation, via scripts in command line mode. The scripts enable multiple simulation runs and automatic generation of means and standard deviation values over all the runs.

Configurable mobility parameters of the simulation model include: the number of nodes in each actual-mobility category {A-ST, A-RWK, A-RWK-G, A-RWP, A-RWP-G}; mean speed of movement; and mean trajectory randomness. For A-RWP mobile nodes the mean duty-cycle between the pause time and periods of movement can be adjusted. Nodes can be grouped in pairs or triples to investigate the performance of the detection of group mobility.

The model is dimensionless, node cell radius is in arbitrary units. This allows any particular technology to be evaluated by attributing some range value to the cell radius unit.

4.2. Evaluation

SDMS is evaluated in terms of its accuracy in determining mobility-status under wide-ranging topological conditions, designed to be broad enough to represent the various mobility models discussed in Section 2.1. In this regard, the sensitivity to five specific important real-world WSN deployment parameters: speed of mobile node movement, the proportion of nodes that are mobile, the neighbour degree, the system scale and the extent of randomness of the trajectories of moving nodes; is investigated.

The algorithm is evaluated in isolation of other services and therefore periodic ($T_{\text{Beacon}} = 1000$ ms) status messages are added to support neighbour table updates. This beaconing rate ensures sufficiently fine-grained operation to reflect the rate of typical application-level events arising from node mobility, whilst avoiding excessive communication overheads.

The first three experiments described in this section are based on systems of 100 nodes. Due to the non-deterministic nature of the target systems; results are based on mean values over thirty simulation runs, each with a new randomly-generated topology.

In graph legends, the symbol ► signifies that the trace shows the mean fraction of nodes with the actual mobility state shown left of the symbol who correctly determine their mobility to be the I-state or T-state shown right of the symbol.

4.2.1. Sensitivity to speed of movement

This experiment examines the impact of dynamically changing neighbourhoods as nodes move in and out of range of each other over a wide range of movement speeds. As the model is dimensionless (in particular the node radius value), the speed parameter simply controls the rate of progressing the node along its trajectory in terms of the number of model timesteps (*NT*) between position updates (recall that the simulation has a very fine-grained 1-ms tick, to prevent simulation-level artefacts from affecting accuracy at the much coarser-grained level of interesting events such as beacon messages at T_{Beacon}). However, the speed parameter can be mapped to real values depending on the transmission range attributed to the cell radius values. (5) relates the speed parameter to the model behaviour:

$$NT = 50 + \big((100 - S) * 10\big) \tag{5}$$

where S is the mean speed of movement parameter, $0 \leqslant S \leqslant 100$. Thus at $S = 0$ the node moves 1 distance unit every 1050 model timesteps, whilst at $S = 100$ the node moves 1 distance unit every 50 model timesteps (for this experiment the node radius was fixed for all nodes at 80 distance units). A random offset in the range $\pm 20\%$ is added to the system-wide value of S for each specific node's speed value.

Fig. 6 shows that algorithm's sensitivity to the mean speed of movement varies for the different mobility classes. The speed values S {0 ... 100} represent the limits of the wide range tested (where 0 means the slowest movement speed tested, but this does not mean that nodes were stationary, see (5)).

The algorithm performs well at detecting independent mobile nodes (RWK and RWP), and at differentiating between stationary and mobile nodes across the full range of tested node movement speeds. However, there is a general improvement in correct mobility-state detection as movement speed is increased. As the detection of mobility is based on detected changes in the local neighbourhood, it is generally more difficult to detect moving nodes at slower speeds. Similarly, at higher speeds, some stationary nodes encounter a high rate of discovery of passing mobile nodes and thus it becomes more difficult to separate out the stationary nodes (it can appear locally that it is the stationary node that is moving past the

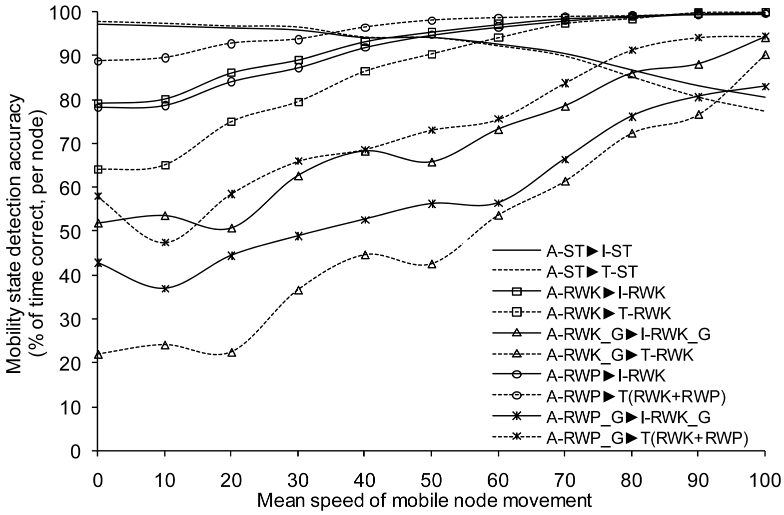

Fig. 6. Performance sensitivity to movement speed.

others in some circumstances). It was found that, due to the pattern of movement of RWP nodes, it can be difficult to distinguish between T-RWP and T-RWK states. At slower speeds the stop–start pattern was well-detected by the metrics collectively, but as node mobility speed increases there is an increasing tendency to classify RWP nodes as having RWK mobility trends and under some circumstances the latter trend can actually be more representative. Thus for RWP nodes the trend predictions of T-RWK and T-RWP can both be considered correct in general, and thus a single combined curve T(RWK+RWP) is plotted (this convention has been followed in *all* of the results presented). The detection of group mobility is harder generally (see the lower four traces), but good performance is achieved at the higher mobility speeds. At slower speeds these moving groups are harder to detect than independently moving nodes, and can be miss-identified as being stationary at times when they temporarily have a higher ratio of long-term known neighbours.

Generally, performance improves as the distance moved between beacon messages increases. Thus improved accuracy is achieved where nodes move relatively faster or where the interval between beacons is longer, which is a good indicator for scalability.

4.2.2. Sensitivity to general level of mobility

This experiment explores the significance of the ratio of mobile nodes (independent and in small groups) versus static nodes, in systems of 100 nodes.

Fig. 7 reveals highly reliable classification of independently mobile nodes (RWK and RWP) which is unaffected by the proportion of mobile nodes (in the wide range investigated). As the proportion of mobile nodes exceeds 40% some static nodes are mistakenly classified as being mobile, as they can detect high numbers of mobile nodes passing through (and hence a high rate of new-neighbour discovery). However, static nodes are still correctly identified 70% of the time when as many as 60% of nodes are mobile. Detection of groups (RWK-G and RWP-G) is found to be more difficult across the range, improving slightly when a larger proportion of nodes are mobile.

4.2.3. Sensitivity to neighbour degree

This experiment with 100-node systems focuses on the effect of changing the packing density of nodes in the topology (measured in terms of the *neighbour degree*; i.e. the mean number of neighbours that nodes have). This impacts on the rate at which new-neighbour discoveries occur when nodes are moving.

We have considered that the number of neighbours of each node needs to grow in accordance with $\Theta(\log n)$ if connectivity is to be maintained (where n is the number of nodes) as shown by Xue and Kumar [25]. This means that the neighbour degree lower and upper bounds for pair connection are $0.074 \log n$ and $5.1774 \log n$ respectively. Taking this result into account, in our investigation the neighbour degree is varied from 2 to 7 (keeping within the connectivity bounds).

Fig. 8 shows that the recognition performance is good over all neighbour-degree values, for all categories of mobility status with general improvement in higher density topologies. This is because more beacon messages are received (from both stationary and mobile neighbours), and thus provides a better information base.

For a given topology, the neighbour degree is dependent on the radio communication range. The issues of heterogeneous communication range and adjustable transmission power are left for further work.

4.2.4. Scalability

The previous experiments were based on 100-node systems. Here we investigate the effect of increasing system size.

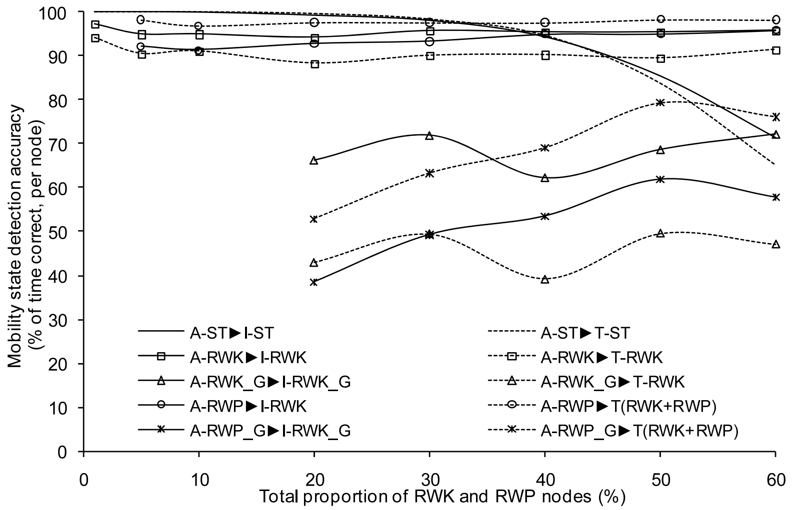

Fig. 7. Sensitivity to the proportion of mobile nodes.

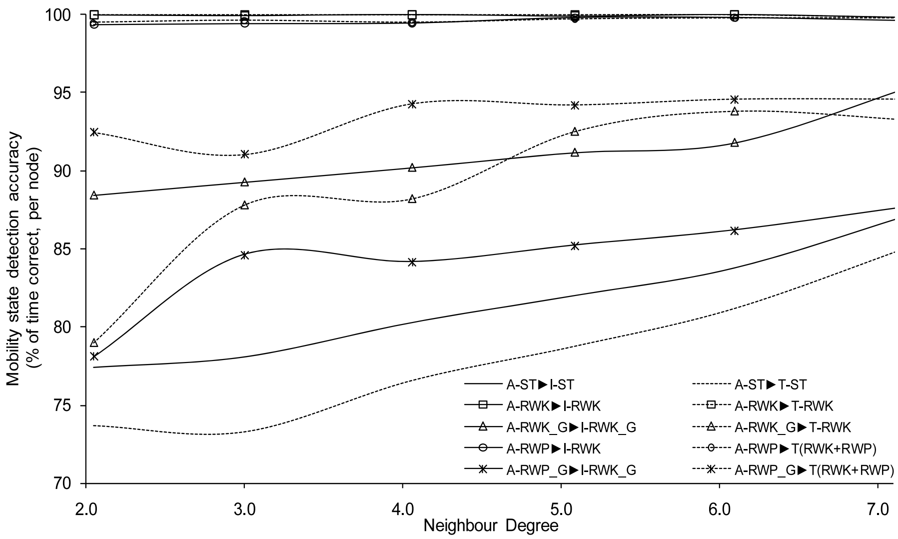

Fig. 8. Sensitivity to neighbour degree.

Fig. 9 reveals that the algorithm performs significantly better at higher scale, for similar reasons that it performs better at higher densities (see Section 4.2.3).

There are noticeable problems detecting static nodes correctly in certain circumstances, more common in smaller systems. The random topologies generated in our experiments can lead to locally sparse areas in which static nodes have few fixed neighbours and thus detect themselves to be mobile when numbers of mobile nodes pass by. This problem occurs less frequently as scale and/or density is increased.

4.2.5. Sensitivity to randomness of trajectory

This experiment investigates the sensitivity of the mobility detection to the extent of randomness in trajectories. Here we take into consideration the fact that the mobility models include the *notion* of randomised trajectory, but do not specify bounds or other characteristics in this regard. We suppose that in a real-world situation it is the *combination* of the underlying mobility model and the specific environment that would determine the actual randomness. For example consider the different patterns that would arise from a tourist loosely following a 'circular city walk' (where the path can be locally linear and where there is a low probability of revisiting the points of interest), contrasted with a criss-crossing path visiting each shoe shop in a given shopping mall several times, comparing products.

Trajectory Randomness (TR) is defined in the model as the rate (in terms of simulation steps) that moving nodes change their trajectories. Performance of the algorithm is evaluated over a wide range of trajectory change intervals. A TR value of 1 represents a short-term constant trajectory (this actually translates into a maximum of 300,000 simulation steps,

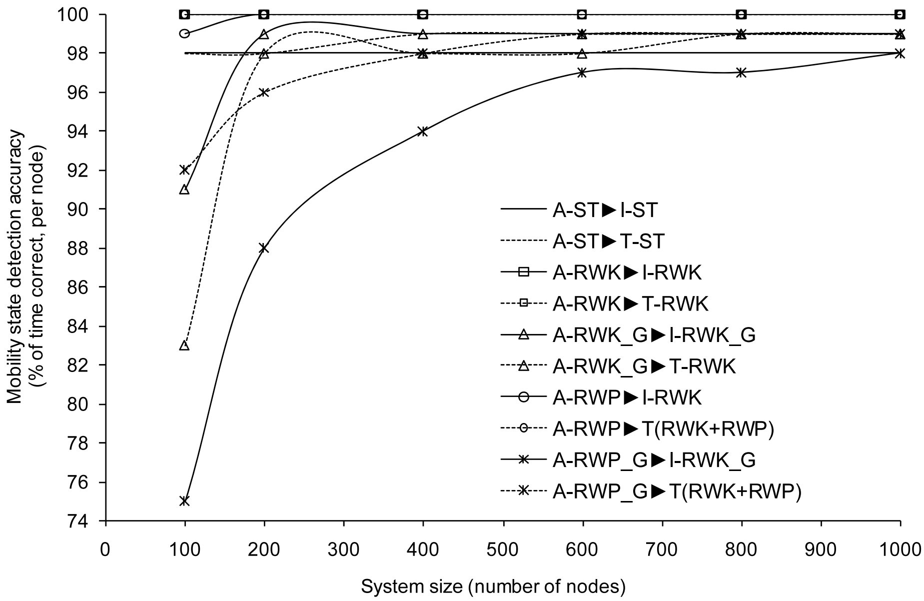

Fig. 9. Sensitivity to system size.

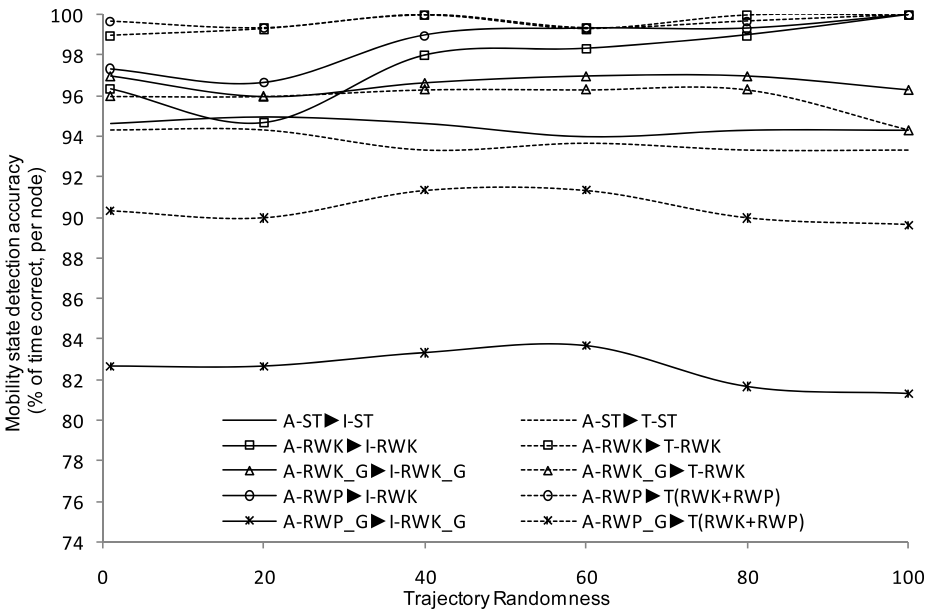

Fig. 10. Sensitivity to trajectory randomness in 1000-node systems with a mean neighbour degree of 7.

equivalent to 300 seconds, in the same trajectory before selecting a new movement direction). A TR value of 100 represents a maximally random trajectory in which the movement follows a fast-changing, seemingly chaotic pattern (this actually translates into selecting a new movement direction as frequently as every 3 seconds). This value range covers all realistic scenarios as well as extremely random and effectively constant trajectories.

For each specific node a random adjustment of up to ±20% is added to the TR change interval to break symmetry and ensure that collective behaviour is realistic.

Fig. 10 indicates that the prediction accuracy for some mobility categories falls slightly as TR increases. Very high values of TR can lead to a node ‘hovering’ about a small area. This could be realistic within a WSN at the scale of a school, hospital or shopping mall for example, where people will visit one general area at a time, and whilst in that area will make small localised movements that may not be detectable by changes in proximity to other WSN nodes using our approach. As a result, the A-RWP-G instantaneous mobility is found to be the hardest to correctly identify; there is a tendency for the nodes to be misclassified as static. However, for many applications this is acceptable because at these times we consider that the people associated with the mobile devices are *effectively* stationary; the logical connectivity does not change if the movement is sufficiently small that the same neighbour devices are maintained (this scenario is aligned with Fig. 2B, Lévy Flights). The trend-level prediction for the A-RWP-G category is much better but still poor relative to the other mobility classes.

The results show that the algorithm is generally insensitive to TR extent, and thus is applicable over a wide range of mobility scenarios.

4.2.6. Summary of results

The algorithm has been evaluated over a wide range of conditions, to stress it and determine the boundaries of its effectiveness. However, it is expected that realistic scenarios form a more restricted subset of circumstances – for example in terms of the ratio of moving to static nodes, as well as in terms of movement speed and trajectory randomness. Within the wider range of values explored, there are combinations of environmental factors in which the algorithm is found to struggle to clearly differentiate between all classes of mobility. However, when the more restricted, realistic scenarios are considered, the algorithm is found to achieve good performance, justified in the sense that the level of differentiation achieved has high value to applications so that they can adjust behaviour accordingly. The performance is achieved using only a small amount of information and at very low cost in terms of communication overhead, processing and storage (as discussed in Section 3.6).

4.3. Comparison with other schemes

The starting point for the current work was the exploration of the use of engineered emergence techniques to self-organise wireless sensor nodes into clusters of a dynamically specified size, based on application requirements [19]. In that work, the concept of a mobile node was introduced to evaluate its effect on the formation of clusters. This investigation led to two extensions to the self-organising cluster technique used: (1) a way to locally detect mobility was introduced, so that each node could dynamically determine whether it was moving or not; and (2) adjustments to increase the beaconing rate are automatically done when either a node detects that it is mobile, or that one of its neighbours is. However, the focus of that work was on the ability to build correct-sized clusters rather than the accuracy of mobility detection, and only a maximum of one node was moving at any time. The current work involves a much more sophisticated mobility detection algorithm, and the investigation is focussed on the mobility detection correctness under different circumstances.

Other schemes in which mobility detection has been investigated include:

MMAC [26] illustrates a useful application of mobility information. It uses detected mobility patterns to adapt nodes transmission rights (dynamic time frame based access) and thus makes the MAC protocol more efficient than a fixed frame time approach when some nodes are mobile. However the protocol assumes that nodes are aware of their location and thus this approach discounts the overhead that such an assumption hides. The technique operates at the MAC layer, as the estimation of mobility state is derived from Received Signal Strength Indicator (RSSI) values or time of arrival (ToA). Detected changes in the two-hop neighbourhood are used to adapt the size of the frames. In [26], MMAC was compared with other MAC protocols: CSMA, TRAMA [27] and S-MAC [28]. MMAC outperformed the other protocols in terms of percentage of packets received and energy efficiency, but it is beaten dramatically (by a factor in the order of 100) in terms of delivery delay by the contention-based CSMA and S-MAC, because of the latency introduced by the random scheduling in MMAC.

MS-MAC [29] is an extension of S-MAC to support mobility. To decrease the time a sensor node needs to join a virtual cluster, a sensor node increases the rate at which it checks for new schedules depending on the estimated movement around the sensor node. To estimate movement, each sensor node records RSSI values for each neighbour and uses any changes as indications of sensor node movement. The calculation of mobility extent is based on the maximum estimated movement speed amongst the neighbours of one node. The frequency of SYNC messages (which are effectively the same as a beacon) will become higher as the calculated mobility speed increases. When the network is static the SYNC messages are sent every 2 minutes and they last for 10 seconds as in S-MAC. Estimated mobility information is included in SYNC messages. Thus the relative speed of one node can impact the two-hop neighbourhood measurement of mobility, also termed the 'active zone'.

MAC feedback carrying information has also been used in link estimation and therefore neighbour discovery. Nonetheless, the use of MAC feedback has mainly been for saving energy in link estimation compared to frequent broadcast beacon exchanges. Some protocols, such as [30], use a combination of broadcast-based and MAC-feedback-based link estimation. They found that overhead is reduced by 25% and utilisation of network capacity is increased by 14%. However, the communication overheads of such an approach are dependent on the wireless technology used, as well as the amount of content transmitted, and also whether the messages are transmitted solely for the mobility estimation purpose, or whether they are more general status messages used to maintain the neighbour table and therefore are not a direct additional cost (this is the situation with SDMS).

While the MAC layer mobility estimation is reported to be more energy efficient in [31,32], broadcast beacons can be used as the basis of link estimation in low-power sensor networks. This is an important consideration because link estimation accuracy significantly affects routing performance. The testbed measurements show that MAC layer link estimation and routing greatly improves the sensed-event reliability (by 18.75%) and energy efficiency (by a factor of 1.96) over purely beacon-based approaches. These results provide empirical evidence of the tradeoff between accuracy and energy efficiency of beacon-based estimation and the MAC layer link estimation.

5. Conclusions

This work has presented a technique for potentially mobile nodes to self-detect their mobility status and pattern in WSN and ubiquitous applications.

WSN sensor nodes tend to be resource constrained. For this reason SDMS has been designed to be highly efficient in terms of communication overhead and computational requirements. This helps to preserve battery power and to enhance scalability by minimising the use of network bandwidth. Efficiency can be further enhanced by integrating the algorithm with some other WSN self-organisation mechanisms, re-using existing communication and data structures. SDMS uses only information from the neighbour table at the node (and thus if this table is already maintained by another protocol in the software stack, SDMS does not impose any additional communication cost). The algorithm is platform independent and does not require any direct negotiation between nodes, or any external information such as GPS data, user input, or hardcoded mobility status (e.g. for initially fixed nodes).

Our work targets systems in which devices' mobility patterns, both immediate and trend, change over time. We consider that in order to support self-organisation, self-optimisation and scalability, the mobility characteristics should be dynamically and locally detected.

We initially chose two of the most generic and commonly applicable mobility models (RWK and RWP) that together represent most real systems to some extent. We have also studied group-mobility variations of each.

The performance of SDMS has been evaluated under a wide range of topological conditions using a sophisticated simulation model. The algorithm is able to differentiate between stationary and independently mobile nodes with a high degree of reliability. The detection of small groups of nodes travelling together has been investigated. This is more difficult because such groups have some characteristics of mobile nodes and some of static nodes. A further scenario that has been investigated is that of nodes having intermittent mobility. This impacts on the value of immediate mobility state for prediction purposes, hence SDMS also provides mobility trend prediction. The combination of groups and intermittent mobility further compounds the detection problem; this is one of several areas of focus for continuing work.

This algorithm fundamentally detects *relative* mobility, so it is most effective in situations where there is mix of moving and stationary nodes, and most valuable where individual nodes movement is otherwise unpredictable but important to detect. The empirical results confirm that the algorithm's performance is indeed correspondingly better in the zone of scenarios in which it is more valuable.

Overall, our experiments show that the accuracy of the approach improves in line with the complexity of the system, i.e. it is better at detecting mobility characteristics when the nodes are moving faster, and also that performance improves in more-dense and larger scale systems. This is very encouraging and we now intend to extend the work to include some more specialised mobility models that relate to specific applications.

This work has shown that it is possible to infer mobility status based only on learnt neighbourhood characteristics typically available from neighbour tables and beacon messages. Information concerning instantaneous and trend mobility state can thus be made available to higher level self-organisation protocols, improving their correctness and efficiency in various application-context specific ways.

6. Further work

The combined state-space of mobility characteristics, self-organising behaviour in networks of wireless and mobile devices, and the requirements and behaviour of ubiquitous applications is so wide that it is not possible to capture all aspects in a single investigation. There are many aspects of mobility and of its detection, characterisation and use as context to drive self-optimisation that are beyond the scope of this paper.

With respect to models of mobility, a main area of interest for us is to investigate the extent to which Lévy Flights (as well as the other models) can be used to describe and predict device mobility in WSNs and ubiquitous applications within bounded places such as factories, warehouses, hospitals, etc. In particular we want to find out whether the well-known models that have been explored at the level of cities also apply at the level of WSNs.

We are also interested in limitations of the well-known mobility models themselves: in particular we are interested in the possibility of hybrid models that are more representative of human movement; this is particularly relevant for ubiquitous computing and pervasive applications. While we have considered random walk and random waypoint entity and group mobility models for our analysis; in real-world scenarios it is rare that groups of people are located in completely unobstructed areas. Additionally, it is unlikely to be the case that people follow random trajectories. In this regard we investigated how our technique performs under a range of different trajectory randomness conditions and we intend to continue this investigation further. We plan to test the accuracy of the proposed algorithm with application specific mobility scenarios.

An assumption in our current model, as with most other models of WSN, is that of homogeneous sensor nodes with respect to radio communication range. This is realistic for typical current WSN deployments, but will be less so in the future. Adjustable power wireless modules are becoming more commonplace and their increasing popularity is driven in part by the need for smart power management. We intend to investigate the impact of heterogeneous radio transmission power on our algorithm's performance.

References

[1] A. Labrinidis, A. Stefanidis, Panel on mobility in sensor networks, in: Proc. 6th Intl. Conf. Mobile Data Management (MDM'05), May 2005, pp. 333–334.
[2] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, D. Rubenstein, Energy efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet, in: Proc. 10th Intl. Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS-X), San Jose, CA, October 2002.
[3] E. Giordano, A. Ghosh, G. Pau, M. Gerla, Experimental evaluation of peer to peer applications in vehicular ad-hoc networks, in: Proc. 1st Annual Intl. Symp. Vehicular Comp. Systems, Dublin, Ireland, July 2008, ICST, Brussels.
[4] J. Cheng, S. Ruan, R. Cheng, T. Hsu, PADCP: Power-aware dynamic clustering protocol for wireless sensor networks, in: IFIP Intl. Conf. Wireless and Optical Comms. Networks, 11–13 April 2006.
[5] E. Ekici, Y. Gu, D. Bozdag, Mobility-based communication in wireless sensor networks, IEEE Commun. Mag. 44 (7) (July 2006) 56–62.
[6] T. Camp, J. Boleng, V. Davies, A survey of mobility models for ad hoc network research, in: Special Issue on Mobile Ad Hoc Networking Research, Trends and Applications, Wireless Commun. Mobile Comput. 2 (5) (2002) 483–502.
[7] J. Yoon, M. Liu, B. Noble, Random waypoint considered harmful, in: IEEE INFOCOM 2003.
[8] M. Musolesi, C. Mascolo, A community based mobility model for ad hoc networks research, in: REALMAN'06, Florence, Italy, May 26, 2006.
[9] Lévy Flights, Wikipedia, http://en.wikipedia.org/wiki/L%C3%A9vy_flight.
[10] M.C. González, C.A. Hidalgo, A. Barabási, Understanding individual human mobility patterns, Nature 453 (5 June 2008) 779–782 (letter).
[11] M. Ghassemian, H. Aghvami, Mobility to improve efficiency of data collection in wireless sensor networks, in: IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, September 2008.
[12] P. Hui, J. Leguay, J. Crowcroft, J. Scott, T. Friedman, V. Conan, Osmosis in pocket switched networks, in: Proc. 1st Intl. Conf. Communications and Networking in China (CHINACOM 2006), Beijing, China, October 2006.
[13] P. Cheng, J. Weng, L. Tung, K. Lee, M. Gerla, J. Härriy, GeoDTN+Nav: A hybrid geographic and DTN routing with navigation assistance in urban vehicular networks, in: Proc. 1st Annual Intl. Symp. Vehicular Computing Systems, Dublin, Ireland, July 2008, ICST, Brussels.
[14] M. Huebscher, J. McCann, A. Hoskins, Context as autonomic intelligence in a ubiquitous computing environment, Int. J. Internet Protocol Technol. 2 (1) (2007) 30–39.
[15] A. Papadopoulous, A. Univaq, J. McCann, Connectionless probabilistic (CoP) routing: An efficient protocol for mobile wireless ad-hoc sensor, in: 24th IEEE International Performance Computing and Communications Conference IPCCC 2005, April 7–9, 2005, Phoenix, Arizona.
[16] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, in: Proc. Hawaii Conf. on System Sciences, January 2000.
[17] Y. Guo, J. McNair, Cost efficient cluster formation for wireless sensor networks, in: Proc. of CITSA 2004, Orlando, FL, July 2004.
[18] L. Sha, J.C. Hou, W.-P. Chen, Dynamic clustering for acoustic target tracking in wireless sensor networks, IEEE Trans. Mobile Comput. 3 (3) (July 2004) 258–271.
[19] R. Anthony, J. McCann, Emergent self-organisation of wireless sensor networks, in: 2nd Intl. Workshop on Engineering Emergence in Decentralised Autonomic Systems (EEDAS 2007), Jacksonville, FL, USA, June 2007, CMS Press, 2007, pp. 2–11.
[20] M. Mostafavi, M. Ghassemian, V. Friderikos, A.H. Aghvami, On mobility metrics applied for ad hoc network protocol evaluation, in: The 7th IFIP International Conference on Mobile and Wireless Communications Networks (MWCN), Marrakech, Morocco, September 2005.
[21] A. Nayebi, G. Karlsson, Beaconing in wireless mobile networks, in: IEEE WCNC 2009.
[22] R. Anthony, Scalable and efficient graph colouring in 3 dimensions using emergence engineering principles, in: 2nd IEEE Intl. Conf. Self-Adaptive and Self-Organising Systems, Venice, Italy, October 2008, IEEE Computer Society, 2008, pp. 370–379.
[23] R. Zoican, D. Galatchil, Mobility in hybrid networks architectures, in: TELSIKS 2005, Serbia and Montenegro, 2005.
[24] SDMS simulator available at: http://staffweb.cms.gre.ac.uk/~ar26/Research/CurrentResearch/SelfDetectMobilityStatus/SelfDetectMobilityStatus.htm.
[25] F. Xue, P.R. Kumar, The number of neighbors needed for connectivity of wireless networks, Wireless Networks 10 (2) (2004) 169–181.
[26] M. Ali, T. Suleman, Z.A. Uzmi, MMAC: A mobility-adaptive, collision-free MAC protocol for wireless sensor networks, in: Proc. 24th IEEE IPCCC'05, Phoenix, AZ, USA, April 2005.
[27] V. Rajendran, K. Obraczka, J.J. Garcia-Luna-Aceves, Energy-efficient collision-free medium access control for wireless sensor networks, in: ACM SenSys 2003, USA.
[28] W. Ye, J. Heidemann, D. Estrin, An energy efficient MAC protocol for wireless sensor networks, in: IEEE INFOCOM, New York, NY, June 2002, pp. 1567–1576.
[29] H. Pham, S. Jha, An adaptive mobility-aware MAC protocol for sensor networks (MS-MAC), in: Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS), 2004, pp. 214–226.
[30] K. Kim, K.G. Shin, On accurate measurement of link quality in multi-hop wireless mesh networks, in: ACM MobiCom, 2006.
[31] H. Zhang, L. Sang, A. Arora, Unraveling the subtleties of link estimation and routing in wireless sensor networks, in: SIGCOMM'08, Seattle, WA, USA, August 17–22, 2008, pp. 473–474.
[32] H. Zhang, L. Sang, A. Arora, Data-driven link estimation in low-power wireless networks: An accuracy perspective, Technical Report: WSU-CS-DNC-TR-08-02, Wayne State University, USA. Available at: http://www.cs.wayne.edu/~hzhang/group/TR/DNC-TR-08-02.pdf.