Des Autom Embed Syst
DOI 10.1007/s10617-013-9126-1

A self-adaptive fault-tolerant systems for a dependable Wireless Sensor Networks

Tiong Hoo Lim · Iain Bate · Jon Timmis

Received: 5 September 2013 / Accepted: 12 December 2013

Abstract As the size and complexity of Wireless Sensor Networks (WSN) continue to grow, there is a need to develop techniques capable of achieving a level of service with successful operations upon which users can depend on. The routing protocol plays an important role in a multihop WSN as it manages and controls the delivery of the data packets. The function of a WSN can be affected by radio anomalies that may degrade the performance of the network. Unreliable and irregular link qualities, due to interference, are common in WSN as the nodes use the same frequency range as the other radio devices. A self-adaptive fault-tolerant network is required that has ability to maintain the level of service even in the presence of faults. Each node needs to monitor and adapt its routing protocols according to the operating environment. Due to resources constraint in the node, it must be carried out in an energy-efficient way and must be dependable. In this paper, we propose an immune-inspired algorithm that provides a level of "self-healing" in the network, through a combined process of self-detection, self-diagnosis and self-recovery and the Immune-inspired Detection and Recovery Systems (IDRS) is presented. In order to evaluate the performance of IDRS, a trace-based simulation, using traces from the hardware, is proposed to analyse the robustness and scalability. The Systematic Protocol Evaluation Technique (SPET) is applied to measure the dependability of the routing protocol. The proposed solution immune-inspired solution using multi-modal mechanism has achieved a higher dependability than existing re-

T.H. Lim (✉) · I. Bate
Department of Computer Science, Deramore Lane, University of York, Heslington, York, YO10 5GH, UK
e-mail: thlim@brunet.bn

I. Bate
e-mail: iain.bate@york.ac.uk

T.H. Lim
Electrical and Electronics Engineering, Institut Teknologi Brunei, Gadong BE 1410, Negara Brunei Darussalam

J. Timmis
Department of Electronics, University of York, Heslington, York, YO10 5DD, UK
e-mail: jon.timmis@york.ac.uk

Published online: 14 January 2014

active routing approaches and can adapt to the current operating environment to achieve the level of service required. Both the hardware and simulation results have validated the accuracy and the performance of the proposed systems. The simulated results have demonstrated that the IDRS can be scaled to a larger networks.

Keywords Wireless Sensor Networks · Self-adaptive routing · Self-healing · Artificial immune systems · Fault tolerance · Dependability · Multi-modal · Trace-based simulation

Abbreviations

WSN: Wireless Sensor Network
MRP: Multimodal Routing Protocol
RSSI: Radio Signal Strength Indicator
PSR: Packet Sending Ratio
PRR: Packet Reception Rate
RDM: RDA Diagnostic Module
NST: Not-So-Tiny AODV
TPC: Transmission Power Control
PDR: Packet Delivery Rate
RT: Retransmission
GD: Global Discovery
SPET: Systematic Protocol Evaluation Technique
RDA: Receptor Density Algorithm
FFT: Fast Fourier Transform
TCR: T-Cell Receptor
MDM: MRP Detection Module
RIRM: Radio Interference Response Module
MRP: Multimodal Routing Protocol
MTPC: MRP Transmission Power Control
TO: Transmission Overhead
LD: Local Discovery
RFI: Radio Frequency Interference

1 Introduction

With the adoption of WSNs in safety critical systems, it is not only sufficient for WSNs to maintain operation for a long period of time with best effort delivery service, they must to provide a reliable end-to-end packet delivery for the application to maintain operation. This is usually achievable if the wireless sensor nodes, also known as the motes, have the knowledge of its current network condition and the control of the protocols operating. Most importantly is the ability to self-adapt the operation to changes in the environment and to *self-heal* in the presence of component or network failures. We define *Self-heal* as the ability to maintain the service through some form of reconfiguration.

With the use of wireless channels, the radio and the packet data can be exposed and subjected to anomalies [15]. Anomalies are *observations that do not correspond to a well defined notion of normal behaviours* [5]. *Anomalies* in WSNs can be caused by a *fault* in the network. The *fault* if not detected, identified and fixed may lead to network *failure* or even network outage, affecting the dependability of the WSNs. The fact that the nodes share the same radio frequency spectrum used by other wireless devices cannot always guarantee

a reliable communication, making them vulnerable to signal distortion, ambient noise and interference. Studies have shown that interferences created by these devices can disrupt the normal routing operation of the sensor node leading to significant packet losses and delays [2]. As reliable communication is an important aspect in safety critical systems, a dependable protocol is required that is resistant to the network anomalies related to radio interference.

Despite various success stories of WSNs in test environments, evidence from previous deployments have shown unacceptable levels of reliability [35]. This can be caused by error in the simulation, testing procedure or the communication protocols. There is a need to improve the reliability and efficiency of the WSNs protocols in order to operate in a dynamic and changing environment. To achieve sufficient level of dependability in WSNs, different fault prevention and recovery solutions have been proposed in the WSN literature to tolerate failure either through the development of new protocol or enhancement of existing protocol. Most of these works have focused on reactive routing due to its energy saving on-demand ability to discover a new route when multihop communications between two nodes are required. In order to provide an appropriate routing strategy, the node must be able to identify and rectify the failure quickly and reliably. It is necessary to build a fault-tolerant network that has the ability to adapt to the current operating environment in order to deliver the required service in the presence of faults.

Designing a fault-tolerant network for WSNs is a challenging task due to the limited resources in the nodes. Most fault-tolerant approaches applied in WSNs demand high resources and require offline regeneration of the detection model. Some researchers attempt to improve the network availability and reliability through redundancy [42], or by detecting the fault with limited automated recovery [5]. It is sometimes necessary to determine the cause of the fault online and rectify the problem quickly and effectively to reduce networks downtime [4]. As a result, researchers have applied the bio-inspired approaches such as the immune-inspired algorithms to solve complex engineering problems using one or two features of the immune system derived from an analogy of the application process and the biological principles [11]. The immune system has many attractive properties that is similar to WSNs such as self-adaptive, self-healing, robust, scalable, autonomous, and self- stabilisation that can be applied to resolve complex problem such as fault detection [38]. The ability of the immune cells to self-detection, self-identification and self-recovery has provided an inspiration for the application of the immune-inspired algorithm to provide self-healing in WSN.

In this paper, we propose an immune-inspired Interference Detection and Recovery System (IDRS) that allows individual nodes to detect, diagnose and recover from network failure due to radio interferences. The IDRS applies the Multimodal Routing Protocol (MRP) [19] to detect the failure and trigger an immune-inspired algorithm, the Receptor Density Algorithm (RDA) [29] to identify the fault. The RDA is applied due to its autonomous ability to detect and identify anomalies in a dynamic changing environment [29]. Based on the stimulation between MRP and RDA to provide an appropriate response to rectify the failure, each node will be able to recover from the failure autonomously. As a result, a self healing property emerges within the network.

This paper is organised as follows. Section 2 provides the motivation of the work followed by a discussion of why existing approaches are not suitable to distinguish interferences in WSNs in Sect. 3. The RDA is introduced in Sect. 4 follow by the descriptions of the proposed IDRS in Sect. 5. The accuracy, efficiency and reliability of IDRS are evaluated and discussed in Sect. 6. Section 7 ends the paper with a summary and some conclusions.

2 Motivation

Network failures such as unreliable link and packet dropped are common in WSNs as the node shares the same radio frequency with other radio emitting devices such as laptop, tablet and game console. The radio on a node is sensitive to the interference generated by these devices. It is necessary to scan the radio channel for any abnormal radio signal strength that is higher than its current radio signal and avoid transmission during interference. This function is usually provided by the link layer [10]. However, this function only detects the presence of interference but does not determine the characteristics of the interference required for an effective recovery [14]. The level of interference is usually dependent on both the *strength* and *duration* of Radio Frequency Interference (RFI) [23]. It is easy to determine the *strength* of the interference (strong or weak) by reading the Radio Signal Strength Indicator (RSSI) from the radio hardware interface. As these interferences depend on the device usage pattern, it can be a challenge to determine the interference's *duration* that can be irregular and derive a pattern that can be used by the protocol to identify the interference characteristics and apply an appropriate error recovery or avoidance actions. The state of the faulty system must be collected and analysed as soon as the fault occurs for immediate recovery. However, only limited historical data can be stored in the memory for error processing. Diagnostic information is usually sent to the sink for analysis. Due to unreliable communication, this information may be lost during delivery. As a result, there is a need to perform online fault diagnosis on the node based on the alert generated by a fault detection system. The importance of integrating the detection, diagnosis and recovery into one system has been widely ignored in literature as each fault-tolerant approach is usually performed and analysed in isolation [33]. In order to provide a reliable network with minimal disruption, it is necessary to perform fault detection, diagnosis and recovery online in the node locally.

The contributions of this paper are:

1. A systematic design and evaluation of a novel self-adaptive algorithm that is able to accurately detect, identify and adapt the responses accordingly to rectify the fault.
2. A trace-based solution to evaluate the scalability of the IDRS using the SPET (Systematic Protocol Evaluation Technique) [20].

3 Existing approaches

Different route detection and recovery approaches have been proposed such as to retransmit [9], or vary its transmission power in order to communicate with the neighbour [21]. In severe cases, the node may need to establish a new route in order to send the packet [30]. Some of these recovery approaches, such as flooding, are more expensive to execute and time consuming than others and are only effective if they are applied according the interference's patterns [13]. Incorrect response can aggravate a congested network. For example, retransmission is best applied when the interference is temporary and transient. Retransmission can improve the probability of successful transmission and improve the reliability of the network [1]. Local discovery should be avoided in a noisy network as it may lead to a broadcast storm [13]. Hence, it is not only important to detect the presence of an anomaly, but also the cause of an anomaly needs to establish in order to make accurate and automated recovery decision and improve availability.

To perform error detection, diagnosis and recovery in a node can be difficult and expensive. Due to limited constraints in the node, the diagnostic and recovery mechanism taken

should be performed online. It should also be low cost and low overhead with a high probability of rectifying the problem. Zacharias et al. [41] propose the use of signal processing approaches such as the Fast Fourier Transform (FFT) to distinguish the radio interference with a known fixed frequency such as Microwave from Bluetooth and Wireless Local Area Network (WLAN). However, the FFT is only performed in a computer using the RSSI collected from the sensor nodes via the base station as FFT is a computational expensive operation to be performed in the motes. For an n elements series, the FFT algorithm computational complexity is $O(n^2)$ [36]. Ong et al. [28] shows that the execution time can take from 20.39 ms to 166.68 ms in a MSP430 microcontroller depending on the number of operations performed and the input representation (an integer or floating point). With the stringent energy budget in sensor mote, a simple and quick algorithm over complex algorithms is more favourable to reduce execution time and energy consumption.

Existing works mainly focus on anomalies generated by malicious attacks that usually have a static distinctive feature that can easily be classified [26]. Little work has investigated anomalies due to the presence of interferences in the operating environment such as WLAN communication. One of the challenges in detecting anomalies due to these interferences is that the duration and occurrence for these types of anomalies are unpredictable and varies with time [23]. The duration and occurrence of these interferences are usually dependent on the type of radio devices or applications, and their usage pattern. Thus, it can be very difficult to be detected and classified using existing conventional statistical approaches [2].

Lin et al. [22] classifies these interferences into three distinct patterns namely: (i) small fluctuation created by multi-path fading of wireless signals; (ii) large disturbance due to shadowing effect of the presence of obstacles; (iii) continuous large fluctuations caused by WLAN devices. Each of these interference patterns can have different detrimental effects on the Packet Sending Ratio (PSR). Recent work by Wang et al. [39] has shown that the interference from a WLAN network can produce up to 30 % packet losses in WSNs. Hence, it is necessary to collect interference patterns and classify the interference according to the strength and duration namely: weak and short, weak and long, strong and short, strong and long. From the classification, appropriate response to rectify the fault generated by each interference type can be mapped onto each interference class and the detection system can use these classification information to rectify the failure.

4 The immune-inspired solution

The immune system is a unique complex defence system that has the capabilities to learn new disease, recognise previously disease and adapt the system to the new environment. It can identify and eliminate specific organism invading the body. These organisms can be harmful and may trigger an immune response when expose to the immune cells. To defend and protect the body, the immune system uses a multi-layer approaches [25]. Over the years, many immune-inspired algorithms have been applied to different application areas of WSNs such as data classification, anomaly detection and coverage problem. This is partly motivated by the analogy between the characteristics of WSNs and the immune system. Davoudani et al. [7] has provided a mapping between the immune system and the WSNs and have highlighted that the functionality challenges faced by WSNs are similar to those faced by immune systems where each node (a cell in immune system) needs to maintain and regulate its operation as long as possible to meet the application requirements.

4.1 The RDA: Receptor Density Algorithm

In this work, the RDA has been employed to perform the fault detection and identification in order to assist in recovery. The RDA is inspired and developed from models and the interactions of a single immune cell (The T-Cells) and its ability to discriminate between the good and the bad cells using the signals received on its surface sensors (Receptors). This paper does not expand on the biological background of the algorithm or comparative studies with other algorithms: [29] for models of the biology. RDA is a suitable for our problem as it has the ability to learn and discriminate between the normal and abnormal autonomously in a dynamic environment. It has also been shown to yield a high accuracy rate when apply to detect anomalies [29].

4.1.1 The algorithm

The algorithm begins by defining the term *receptor* taken from [29].

Definition A receptor r is a tuple (p, n, β, ℓ, c), where:

$p \in [0, \ell]$, the receptor position;
$n \geq 0$, the generated negative feedback;
$\beta > 0$, the base negative feedback barrier;
$\ell \in (0, \infty)$, $\ell > \beta$, the length of the receptor;
$c = \{0, 1\}$, the receptor output. $c = 1$ if $p \geq \ell$.

This definition of a receptor is *an abstraction of the internal component of the TCR* in which p represents the kinetic proofreading state of the internal component of the TCR and ℓ is the maximum kinetic proofreading state that is capable to generate an activation signal. c is the output that determines whether the T-cell is activated. n represents the generation of negative feedback in the neighbourhood of the receptor and the threshold β is the base negative feedback barrier [29].

The behaviour of the receptor leading to activation is shown in Fig. 1 and is described as follows:

- A receptor receives a sequence of input $\{u_t\}$ with $u_t \in \mathbb{R}$ at discrete time $t = 0, 1, 2, \ldots$ and $u_t \geq 0$.
- The input u_t pushes p_t towards ℓ.
- The receptor generates a negative feedback n_t if the receptor position $p_t \geq \beta$ the negative barrier.
- The negative feedback n_t reverses the progress of the receptor position p_t.
- The receptor position p_t and negative feedback n_t are updated according to a decay function

$$f_t : (p_t, n_t, u_t, \beta) \rightarrow (p_{t+1}, n_{t+1}) \tag{1}$$

where p_{t+1} and n_{t+1} are given in Eq. (2) and Eq. (3).

$$p_{t+1} = bp_t + u_t - an_t \tag{2}$$

$$n_{t+1} = \begin{cases} dn_t & \text{if } p_t < \beta \\ dn_t + g & \text{if } p_t \geq \beta \end{cases} \tag{3}$$

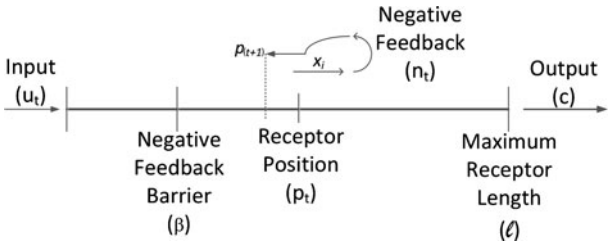


Fig. 1 The Receptor from [29]: An input u_t will move the receptor position to p_t as well the generate a negative feedback if $p_t > \beta$. This negative signal reverses the receptor position to $p(t+1)$. If the subsequent input signal is strong, the p_t progresses further toward ℓ where the output c is produced when $p_t = \ell$

The parameters b and d are the receptor position decay rate and negative feedback decay rate with $0 < b < d < 1$. $a > 0$ controls the influence of negative feedback. $g > 0$ is the negative feedback growth rate.

- If the receptor position is on and above the receptor length ℓ, then a classification occurs and the receptor is activated and considered anomalous ($c = 1$ if $p_t \geq \ell$, otherwise $c = 0$).

5 IDRS: Interference Detection and Recovery Systems

In this section, we propose the application of MRP and RDA to provide self-healing using the current radio RSSI values collected within a time window. Using the multi-layer approaches inspired from the immune systems, the MRP [19] provides the first line of defence to detect the failure and can activate the RDA to identify the fault. The MRP is a multi-modal protocol that has the ability to adapt the routing operation according to the current operating environment. If the MRP fails to improve the network condition, RDA is activated. RDA is designed to detect any continuous time-series anomalous event and has the ability to adapt to dynamic changing environment by reshaping the normal pattern in the system. The RDA uses the RSSI to identify the type of interferences as the RSSI has been recognised as a good predictor of link quality and has been used in routing protocol to assist in detecting link failure [34]. Specifically, it has been shown that if the RSSI is higher than a sensitivity threshold $RSSI_{th}(x)$ (about -87 dBm), the RSSI correlates very well with the Packet Reception Rate (PRR). This will allow us relate the distribution of the RSSI observed to the PRR generated by a MRP for detection and classification and map the results to the response accordingly. However, the raw RSSI values read by the WSN's node consist of a mixture of different signal including noise. It is necessary to process and extract the interference pattern from the raw RSSI. Hence, we integrate the RDA with the MRP to provide an immune inspired self healing system that allow individual node to self-detect, self-diagnose and self-recover from failure online.

The objectives of the IDRS are:

1. To accurately detect and identify the interference that is affecting the communication between a node and its neighbour in a distributed manner,
2. To make autonomous decision on the recovery action to mitigate the effect of the interference, and improve the network reliability and efficiency.

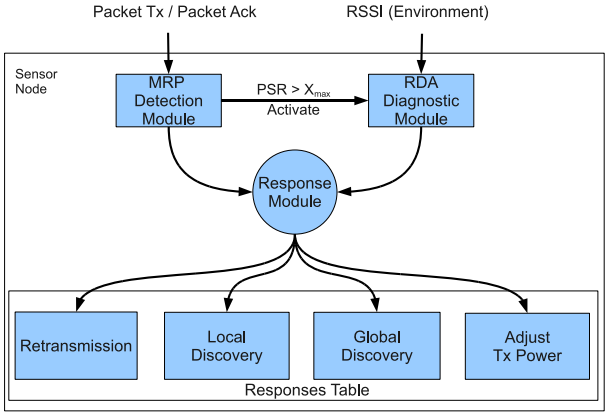


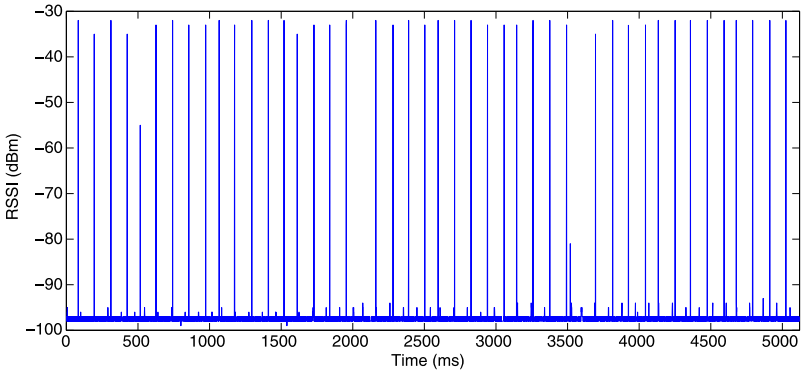
Fig. 2 The architecture of the CIS-based Interference Detection and Recovery System

The IDRS in Fig. 2 consists of three modules, representing each stage of the self-healing process: MRP Detection Module (MDM), RDA Diagnostic Module (RDM), and Radio Interference Response Module (RIRM). Inputs to the IDRS are the PSR and the RSSI. These inputs can be obtained and calculated from the hardware of the node. The MDM acts as the first line of defence to provide initial detection and respond to the interference. The MDM will perform a self-monitor using the PSR to evaluate the outcome of its action. If the PSR does not improve, the MDM will activate the RDM to identify the type of interference based on the current RSSI value. Based on the results from both the MDM and the RDM, the RIRM will activate one or a combination of responses. By using the close feedback loop provided by the link layer, the effectiveness of the responses can be monitored by the MRP and the cost of each response can be adjusted accordingly. As a result, the operation of the node can be adapted to its current operating environment in order to maintain or improve the network service. Hence, the IDRS should be able to recognise and self-adapt respond to the failure based on the strength and duration of the interference.

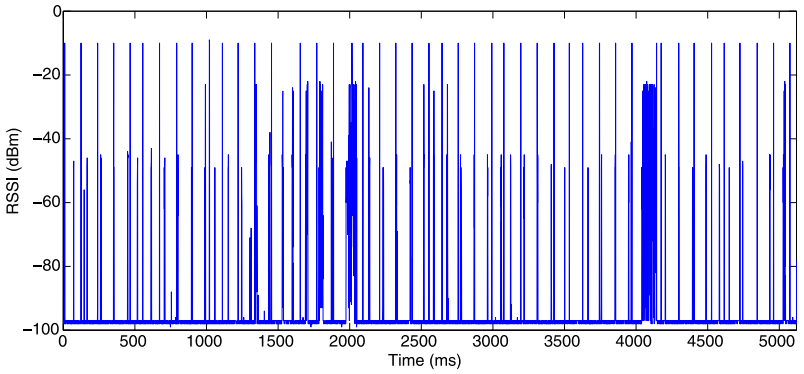
In the following subsections, a detailed description of the proposed IDRS Algorithm is presented.

5.1 MDM: the MRP detection module

In WSNs, the packet reception ratio (PRR) is a commonly used as a metric to detect network anomalies. The PRR is shared between neighbouring nodes [21]. This data is usually piggybacked on an existing packet. However, in the presence of interference this data may be lost or corrupted. Hence, we advocate that the detection module should be implemented at the transmitting node. In the IDRS, we propose the use of the MRP to detect the presence of interference based on the PSR to provide an initial response [19]. The PSR is the total number of packets successfully sent over the total number of attempts made in a given time window. The MRP utilises the packet acknowledgement (P_{ack}) to detect deviation in the PSR, provide initial recovery response and activate the RDM if required. Each route recovery response incurs a specific cost (RT_{cost} for *retransmission*, LD_{cost} for *local recovery*). Associated with each recovery response is a maximum cost threshold: RT_{max} for retransmission and LD_{max} for recovery. The recovery response will only be selected if the cost of carrying out the response is lower than the maximum threshold. All these responses utilise the existing acknowledgement mechanisms on the link layer. As such, no additional communication overhead is incurred in the network.



(a) Normal RSSI Radio Data Pattern



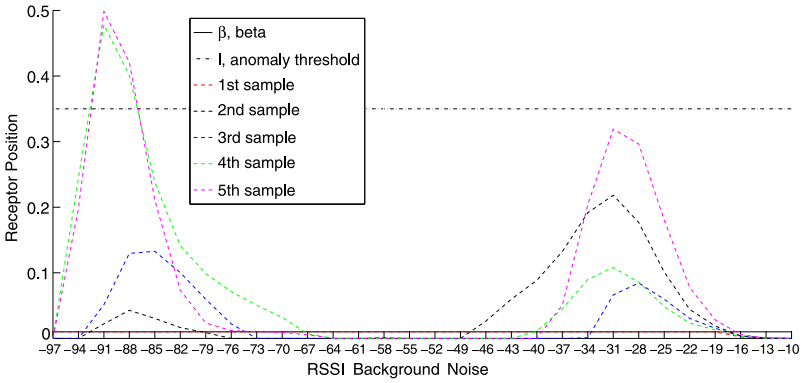
(b) Abnormal RSSI Radio Interference Pattern

Fig. 3 The raw RSSI data collected from the radio interface of a TelosB for both normal (**a**) and abnormal (**b**)

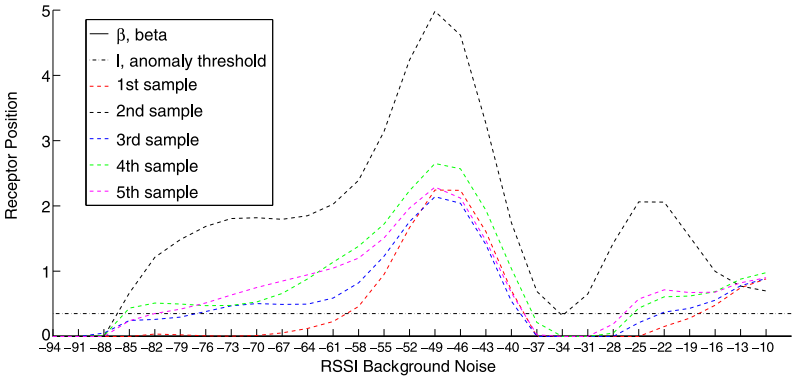
5.2 RDM: the RDA diagnostic module

To identify the cause of a transmission failure, the Received Signal Strength Indicator (RSSI) is used. Monitoring the RSSI in WSNs has been widely used to decide the required transmission power and protocol to transmit a packet [8]. However, as illustrated in Fig. 3, the RSSI values are sensitive to changes in environment. Classifying the RSSI values using traditional statistical techniques to differentiate fluctuating RSSI pattern is challenging [2]. Small changes in the operating environment can trigger large variations in the RSSI, making it difficult to determine the type of interferences [16]. We propose the use of the RDA [29] to filter the background noise and classify the interference. The RDA has ability to achieve high positive detection rate and low false detection rate autonomously [12, 17]. Its ability to recognise anomalies in a dynamic environment has motivated its application to our solution.

To apply the RDA, the RSSI input data is divided into s discretised locations and a *receptor* $\mathbf{x}_s$ is placed at each of these locations. A receptor has a maximum length $\ell = (\sqrt{2\pi})^{-1}$, a position $r_p \in [0, \ell]$, and a negative feedback $r_n \in (0, \ell)$. The maximum receptor length ℓ (activation threshold) is set based on the spread of the RSSI value obtained during training. At each time step t, each receptor takes input $\mathbf{x}_i$ and performs a binary classification $c_t \in 0, 1$ to determine whether that location is considered anomalous. The classification decision is determined by the dynamics of r_p and negative feedback $r_n \in (0, \ell)$.



(a) The signature of activated receptor for the normal RSSI data Pattern



(b) The signature of activated receptor for the abnormal RSSI interference Pattern

Fig. 4 The raw RSSI samples collected over a time window are fed into RDA to product normal (**a**) and abnormal (**b**) signatures of activated receptors

The processes for initialisation and classification of the RSSI values are described as follows:

Phase 1: Initialisation

1. Present the normal RSSI values **X** (Fig. 3a) to the RDA to generate its normal signature (Fig. 4a). For each receptor x , calculate the sum of stimulation $S(x)$ on each receptor x for each RSSI input x_i, $x_i \in \mathbf{X}$.

$$S(x) = \sum_{i=1}^{n} \frac{e^{\frac{-(x-x_i)^2}{2h^2}}}{h\sqrt{2\pi}} \tag{4}$$

where h is the kernel width and n is the total number of normal RSSI values.

It is necessary to use appropriate kernel width to accurately detect the interference. To set the h, three sets of training data (normal, www and video streaming) are used to analyse the effect of h on the detection rate. From Fig. 5, a small $h = 1$ pushes the receptor position far away from the threshold that may increase false positive detection (mis-classification). A large $h = 10$ may decrease the true negative rate (missed detec-

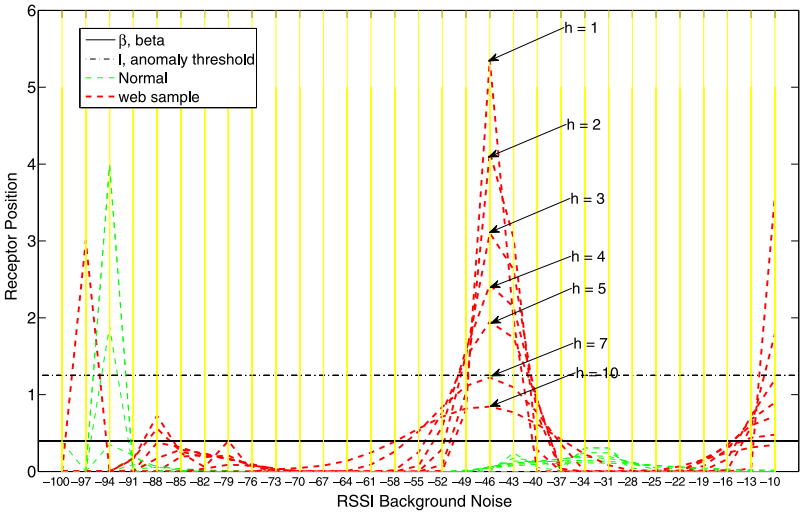


Fig. 5 The signature patterns generated by RDA with the different values h. The receptor position progress faster when the value of h is high and may increase the false positive rate

tion) as the reception position is below the threshold. By manually adjusting the value of h using the training data, the RDA can yield a high true positive rate when $h = 5$.

2. Calculating the negative feedback $r_n(x)$ for each receptor x. $r_n(x)$ slows down the progression of the receptor position to reduce the false positive rate and is computed based on the base negative barrier β. Hence, it is necessary to determine the value of β by adjusting the β using a set of training data. From Fig. 6, the position of the receptors moves toward threshold ℓ faster if β is high (β =1). The progression of the receptor position decreases when β is small ($\beta = 0.01$). By manually inspect the receptor position with different value of β, β is set a low value ($\beta = 0.01$) to yield low false positive rate.

$$r_n(x) = \begin{cases} S(x) - \beta, & \text{if } S(x) \geq \beta \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Phase 2: Classification

1. Initialise the receptor position $r_t(x) = 0$ for all receptors.
2. Based on the MAX($r_p(x)$) of normal signature, set the threshold value of the receptor length $\ell = (\sqrt{2\pi})^{-1}$.
3. Calculate the new receptor position $r_p(x)$ with current RSSI values $\mathbf{V}$.

$$K_s = \sum_{i=1}^{n} \frac{e^{\frac{-(x-v_i)^2}{2h^2}}}{h\sqrt{2\pi}}, \qquad r_p(x) = K_s - r_n(x) \tag{6}$$

where each RSSI value $v_i \in \mathbf{V}$.

4. Classify $\mathbf{V}$:
A receptor is activated when

$$\mathbf{V} = \begin{cases} Normal, & \text{if } r_p(x) < l \\ Interference, & \text{otherwise.} \end{cases} \tag{7}$$

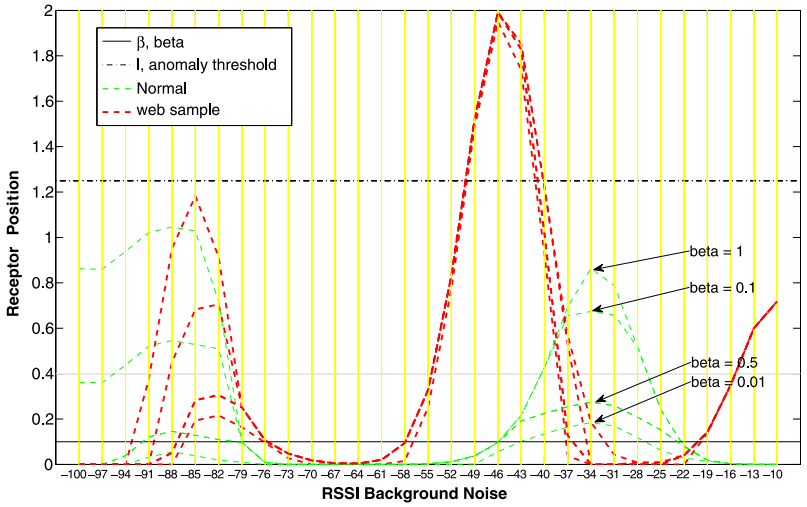


Fig. 6 The signature patterns generated by RDA with the different values β. A low of value β slow down the progress of the receptor position and hence may reduce the false positive rate

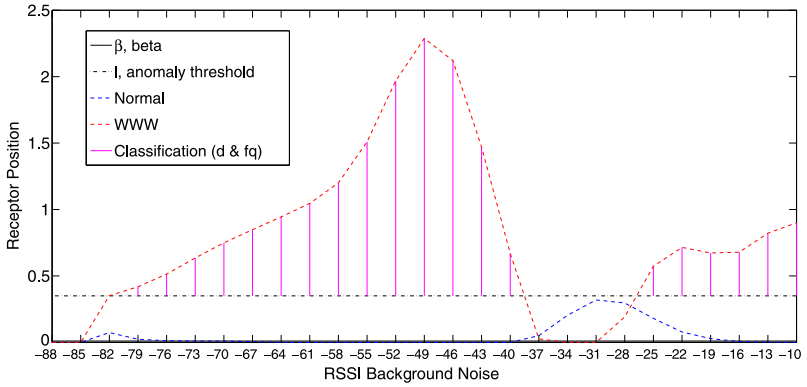


Fig. 7 Using the outputs generated by RDA, the interference can be classified into either Class I, II, or III based on the Euclidean distance of the furthest activated receptor and the number of activated receptors above the threshold l, represented by the global maximum and vertical lines in (c) respectively

The classification of v to different classes of interference is based on two variables (Fig. 7c):

- The difference between distance of the highest receptor position and ℓ $(\max(\ell - r_p(x)))$, referred to as *Intensity*;
- The number of activated receptors, referred to as *Duration*.

5.3 RIRM: Radio Interference Response Module

According to the duration and strength of the interferences, we classify the interferences into four patterns namely: short and weak, short and strong, long and weak, long and strong. Four different responses have been proposed to overcome the four interference patterns:

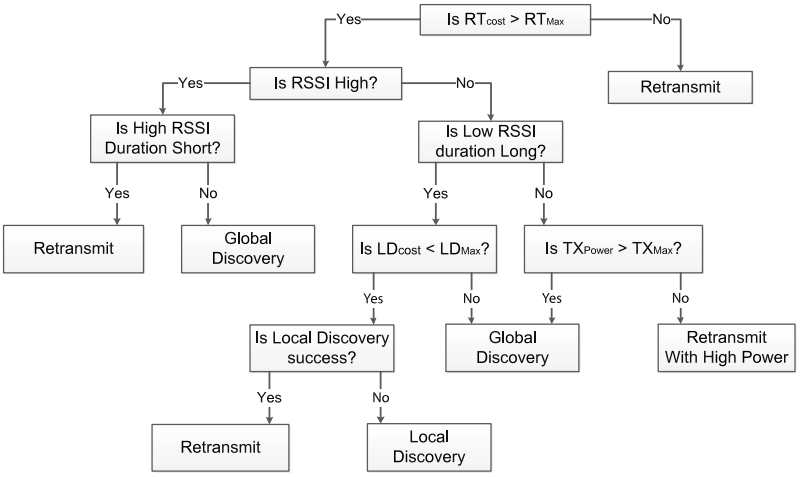


Fig. 8 Decision tree based on expert knowledge for the RIRM and MDM to respond to different interference

1. Short and Strong: Retransmission (RT) is the default action and activated when the MDM detects that the acknowledgement packet P_{ack} is not received and the cost of retransmission RT_{cost} has not exceed the threshold RT_{max}. This response is particularly effective when the interference noise signal is *short* and *strong*. Retransmission is known to improve the reliability if the network is suffering from frequent intermittent packet collisions created by other radio emitting devices with a stronger signal [10].
2. Short and Weak: Increasing the transmission stronger than the interference source can help to handle unavailable route caused by an obstacle or a weak interference source. It is a common to increase the transmission power to penetrate through the obstacle in order to communicate with the next hop neighbouring node [2, 21]. However, the use of higher transmission power can only be applied temporary and when there are no other devices with stronger radio transmission around as it consumes more battery and may also interfere with other nodes. Hence, TPC is only applied when irregular *short* and *weak* interferences are detected or when normal retransmission falls below a threshold and the transmission power Tx_{power} is less than a predefined maximum power Tx_{max}.
3. Long and Weak: Local discovery (LD) is activated when the node failed to send the packet after several RTs (as indicated by PSR) or when the RDM identified an interference that is *long* and *weak*. As a weak interference may only affect one of two nodes, other local nodes within the affected area can still forward the packet to the next two hop neighbour. This response is also best executed when the next node is permanently unavailable.
4. Long and Strong: Global Discovery (GD) is usually the last option to take when the local nodes are spatially interfered and the existing route is known to be unreliable. This action is usually taken when all the previous responses have failed, and there is no local node available to re-route the traffic. This type of failure is usually created by the interference source that is *long* and *strong* affecting all the nodes.

A decision tree, based on expert knowledge and the previous explanation, is presented in Fig. 8 to show the response strategy to be selected based on the current network environment.

Algorithm 1: IDRS Algorithm with the combination of MRP and RDA

```
Input  : Packet Send P_s
Output : Response Action
while Packet Buffer is not Empty do
    Send Packet P_s and wait for acknowledgement P_ack
    if P_ack is not received then
        Calculate Packet Sending Ratio, PSR
    else
        Decrease Retransmission Cost, RT_cost
    if PSR < 95 % then
        Determine interference CLASS from RDM

    if not CLASS III and [PSR > 90 % or RT_cost < RT_max] and Route is valid then
        Retransmit
        Increase Retransmission Cost, RT_cost
    else if CLASS II and LD_cost < LD_max then
        Perform Route Discovery
        Increase Local Discovery Cost LD_cost
        if Route Discovery is Successful then
            Decrease Retransmission Cost RT_cost
    else if CLASS I and Tx_Power < Tx_MAX then
        Increase Transmission power, Tx_power
        Decrease Retransmission Cost, RT_cost
    else
        Invalidate Route and Send Error
        Global Discovery
    if Timeout then
        Reinitialised
```

The algorithm for IDRS with MRP, RDM and RIRM is given in Algorithm 1.

6 Experiments and results

We conducted two experiments to evaluate the proposed IDRS. The first experiment examines the effectiveness of the RDA classifier in the RDM. The second experiment evaluates the efficacy and the scalability of the proposed IDRS when compared to other methods.

6.1 Evaluation of the accuracy of RDM

In the RDM, we use over 850,000 RSSI readings to classify the interference into three classes: CLASS I, CLASS II, and CLASS III. The RSSI values are obtained from the TelosB radio module [31], exposed to different interference sources. The spectrum of the RSSI values used is the range of -100 dBm to -10 dBm. Due to the limited processing power and storage in a sensor node, this spectrum is uniformly divided into 30 slots to ensure that

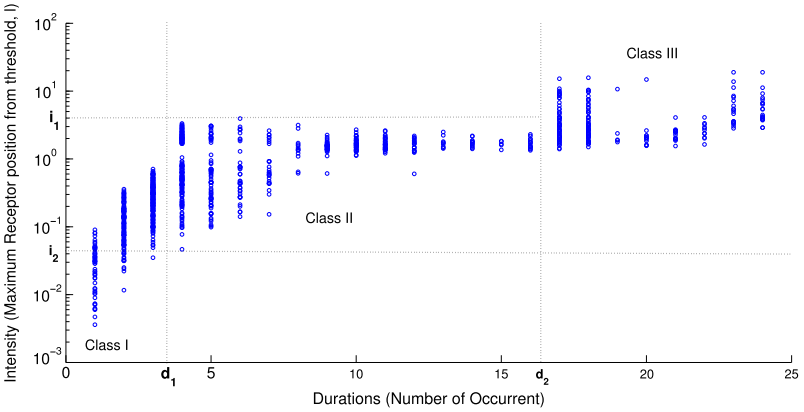


Fig. 9 The distribution of the interference characteristics after processed by the RDA. The distribution can be grouped as the three different interference classes (Class I, II and II) as separated by the *dotted lines*

Table 1 Interference Class based on the intensity and duration of the interference experienced by a node

Intensity, C1	Duration, C2	Class	Remarks
$0 < C1 \leq 2.8$	$0 < C2 \leq 5$	I	weak intensity, short duration
$2.8 < C1 \leq 11.0$	$5 < C2 \leq 16$	II	medium intensity, medium duration
$C1 > 11.0$	$C2 > 16$	III	strong intensity, long duration

each RSSI values are evaluated. The RDA requires $O(n^2)$ processing overhead, hence will only activate if PDR $<$ N (where N is the minimum PDR the application can afford to tolerate). A receptor is used to represent each slot.

In order to classify the interference into different classes, the pre-processed training data from the RDA have to be grouped according the number of activations (duration) and the maximum distance between ℓ and maximum receptor position. A scatter plot is used to investigate whether a general pattern can be observed from the preprocessed data. From Fig. 9, the output generated by the RDA can be grouped into 3 different classes based on different *Intensity* (**C1**) and *Duration* (**C2**) representing:

- **Class I**:- Weak intensity, short duration ($C_1 < i_1$ and $C_2 < d_1$)
- **Class II**:- Medium intensity, medium duration ($i_1 < C_1 < i_2$ and $d_1 < C_2 < d_2$)
- **Class III**:- Strong intensity, long duration ($C_1 > i_2$ and $C_2 > d_2$)

Hence, an unsupervised K-mean clustering algorithm is applied to data group the data into 3 groups. K-means clustering is a method used in data mining to partition v samples into c clusters in which each sample v belongs to the cluster with the nearest mean. It is performed offline as it is computationally expensive. The classes do not usually change throughout the node's lifetime unless it is deployed in a different physical location or moved. The derived classes based on C1 and C2 is shown in Table 1.

6.1.1 Evaluation metrics

We evaluate the performance of the RDM in TelosB mote [31] running TinyOS [18] based on *sensitivity* (8) and *precision* (9).

$$Sensitivity = \frac{TP}{TP + FN} \tag{8}$$

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

where TP (True Positive) = a fault is correctly classified; FP (False Positive) = a normal state is incorrectly classified as a fault; FN (False Negative) = a fault is incorrectly classified as a normal state.

6.1.2 Experimental setup

Two static nodes are deployed so that they are within each other transmission range. One node is configured to transmit packets at the rate of 8 packets per second to simulate heavy load traffic while sampling its radio channel at the rate of 1 kHz to collect the RSSI values and perform online detection.

Eight different network conditions, each representing different interferences commonly occur an office and home environment, are used to test the system namely: normal WSN communication, object blocking, jamming from another node, WLAN traffics such as web browsing (WWW), slow video streaming, fast video streaming, slow file downloading , and fast multiple files downloading (torrent). In each run, the interference is injected into the network at periodic interval to capture the PSR affected by the interference. This is done by placing a laptop next to the receiving node. Due to the limited memory size in the nodes to store the log, each experiment is run for 5 minutes to generate 2400 packets and is repeated 15 times.

6.1.3 Experimental results

The results for the experiment are shown in Fig. 10. From the figure, the RDM has achieved a precision of above 80 % when the interferences have caused a drastic drop in the PSR $<$ 70 %. The occurrence of these two interferences requires an alternative route to deliver the packet successfully. Hence, it is important these two interferences are correctly identified to avoid unnecessary responses to be executed. With RDM, a precision rate from 80 % to 90 % has been achieved for both blocking and fast download. Although the RDM can only classify 50–60 % of the class II interference, its impact on the network PSR is less extreme with more than 80 % of the packet is still being delivered compare to blocking and fast download with the PSR below 70 %. Beside, high accuracy in Class II interference is usually not required for accurate response as the sequential recovery step (Retransmission followed by local discovery) provided by the MRP can usually overcome weak interference and rectify the problem [19].

6.2 Evaluation of the performance of IDRS

In this second experiment two sets of experiments are executed to compare the efficacy and scalability of the IDRS against Not-So-Tiny AODV (NST), MRP, and MRP with adaptive

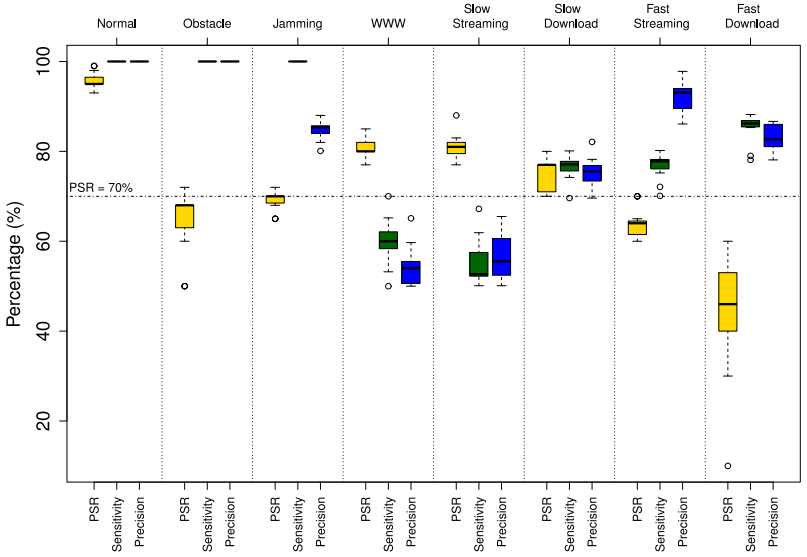


Fig. 10 Results showing the detection accuracy and the effect of different interference classes on PSR. As indicated by PSR = 70 %, the RDM can detect and identify interference that have a severe impact on the PDR as above 80 % precision can be achieved

Transmission Power Control (TPC): one is in hardware and the other in simulation. The objective of the hardware experiment it to show that the IDRS can increase the PDR and reduce the communication overhead in the network. The simulation evaluate the scalability of the IDRS as it is much easier and faster to test a large scale network in simulation than real hardware deployment. With the availability of hardware and simulation, we will also apply the SPET where possible to evaluate and compare the routing protocols in order to achieve confidences in the results. It is worth noting that the IDRS has not been compared against other anomaly tolerance schemes in our experiment because there are not really any similar system to compare against.

To evaluate the performance of the IDRS against the NST, MRP, and MRP with adaptive TPC (MTPC), the routing protocols are installed and tested in both hardware and simulation. The MTPC protocol is used to evaluate the benefit of boosting the transmission power when the receiver is being blocked.

6.2.1 Evaluation metrics

The statistical information collected in a WSNs simulation can be used to show the average performance. The performance of the routing protocols is evaluated based on the following metrics:

– **Packet Delivery Ratio (PDR)**: PDR represent the percentage of the number of packets received by the receiver, to the total number of packets transmitted by the sender. This metric measures the reliability of the routing protocol.
– **Transmission Overhead (TO)**: TO is defined as the average number of transmissions made by a node to deliver the packets to the receiver. This metric represents the effi-

Table 2 The range of A-values proposed by Vargha et al. [37] to represent different effect sizes

Large effect size	Medium effect size	Small effect size
A-value ≤ 0.27	$0.27 < A$-value ≤ 0.36	$0.36 < A$-value < 0.44
A-value ≥ 0.73	$0.64 < A$-value ≤ 0.73	$0.56 < A$-value < 0.64

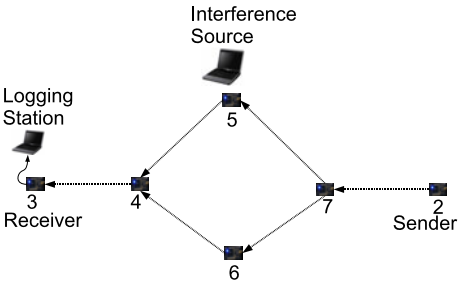


Fig. 11 Interference source is introduced near node 5 and 6 to disrupt the communication between node 2 and 3

ciency of the routing. It can be calculated by dividing the sum of the transmissions made, including RT, LD and GD, to the total number of packets received.

As data generated from experiments can be subject to error. It is necessary to reduce and understand the error by applying statistical analysis to summarise those observations and quantifies the uncertainty in the measured variable. The two statistical tests are used. The first tests evaluate the statistical significant of the results and determine whether protocol X is really better than Y or whether the results are so close than differences are purely random. Mann-Whitney-Wilcoxon test, also known as rank-sum test, is applied to compute the p-value [40]. Based on a pre-determined confidence level of X %, the results are shown to be statistically significant if the p-value $\leq \alpha$. An α value of 0.05 is typically used, corresponding to 95 % confidence levels [40]. The second tests examines the scientific significance of results by measuring the difference or the effect size between the protocols. The Vargha-Delaney A-statistic is used to measure the effect size [37]. The A-value is in the range [0, 1] and is computed using the parameters collected from the previous rank-sum test. Using the guidelines proposed by Vargha et al. [37], the range of A-values representing different effect sizes are presented in Table 2, where the large effect shows that the results are different.

6.2.2 Hardware experimental setup

A small number of (6) nodes were chosen to allow a better greater control of the experiments. The experiments are performed in the centre of a large room relatively free from uncontrolled radio sources, however a larger physical network would then be closer to uncontrolled noise sources. 6 static TelosB motes are placed 3 metres apart using the topology shown in Fig. 11 to ensure that the neighbouring nodes are within each other transmission range. The experiment is conducted at the centre of a room relatively free from uncontrolled radio sources to ensure its correctness and validity. The node transmission is set to minimum power using the same channel as the WLAN in the room. A notebook with different applications will be used as an interference source. The LLN is enabled to allow packet acknowledgement in each node. During initialisation, node 2 is configured to collect temperature reading from the sensor and transmit the packet to node 3, at regular intervals (250 ms) via the intermediate nodes. Once the network route has been established, and the normal signature has been collected by the RDM (after 30 seconds), different interference

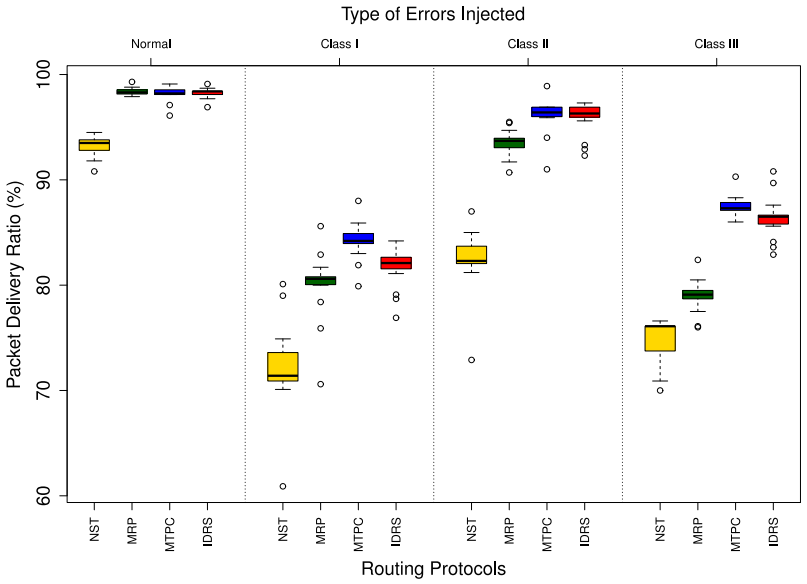


Fig. 12 PDR achieves by different routing protocols for different classes of interference

sources are introduced into the network (close to node 5 and 6) at every i seconds intervals. Each interference lasts for approximately d seconds. In our experiment, the values of i and d are set accordingly ($i = 30$ and $d = 15$) to ensure that the networks can recover before the next interference source is injected. The PDR and the response executed are computed and logged by node 7 during each interference cycle. Due to the limited memory size to store the logs, each experiment is run for 10 minutes to capture the interference sources. Each experiment is repeated 15 times as it takes about 30 minutes to configure and run each experiment in hardware.

Experimental results The results for this experiment are shown in Figs. 12 and 13 with their respective statistical test values in Table 4 and Table 5. During normal condition, NST has the lowest PDR (median at 92.5 %) and requires more transmissions than MRP, TPC and IDRS. The transmission overhead for IDRS is slightly lower than MRP and TPC despite having the same PDR (Rank Sum-p-value < 0.005451, Vargha-Delaney A-value > 0.79778).

When errors are introduced into the network, the performance improvement made by the RDA with MRP is significantly better than MRP as the PDRs for IDRS are always higher than MRP. (For Class I is 2 % higher, Class II is 4 % higher and Class III is 9 % higher with p-value $\ll 0.005$ and A-value < 0.27.) MTPC has the highest PDR in Class I and Class III. Further analysis on the responses executed by the routing protocol in Table 3 has shown the number of TPCs in MTPC is higher than IDRS at class III interference. We believe by increasing the transmission power during interference may have improved the PDR in MTPC. From Fig. 12, by increasing the transmission power during Class I interference has improved the PDR by 5 % on average for MTPC compared to MRP and 2.5 % compared to IDRS.

Although the boxplot in Fig. 12 have shown a significantly higher PDR in MTPC than IDRS , IDRS has a lower number of packet transmissions compare to MTPC as shown in

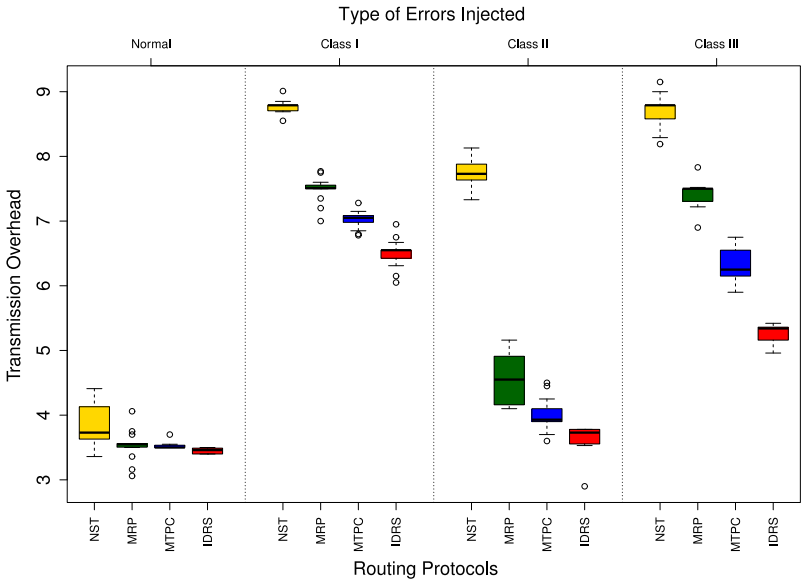


Fig. 13 Transmission Overhead generated by different routing protocols for different classes of interferences

Table 3 The execution of different responses in all the nodes for IDRS and MTPC. Results show that IDRS can execute the appropriate response compared to MTPC. Class III interference has been correctly classified in IDRS resulting in lower number of responses being executed than MTPC (in **Bold**)

Class	Number of responses executed								Interference detected		
	RT		LD		TPC		GD				
	IDRS	MTPC	IDRS	MTPC	IDRS	MTPC	IDRS	MTPC	I	II	III
Normal	41	44	24	35	5	7	50	48	7	0	0
I	113	174	52	92	33	12	342	404	59	4	1
II	42	66	15	14	6	14	88	60	35	15	1
III	**164**	**193**	**108**	**145**	**25**	**44**	**121**	**219**	88	74	82

Fig. 13. When Class I and III errors are introduced, the transmission overhead for IDRS is one less than MTPC although the PDR of MTPC is 2 % higher. From Table 3, we can observe that IDRS generates less packets than MTPC as an appropriate response can be executed by RIRM based on the diagnosis made by the RDM. For example, the IDRS performed less RT and TPC in Class II interference as the node was able to recognise the interference and performed LD immediately (higher LD). The RDM in the IDRS managed to effectively classify the interference as shown in class I, II, and III in Table 3. As a result, the total number of responses performed by IDRS is significantly less than the MTPC. As a result, the IDRS consumes less energy as a lower number of transmissions is required to deliver a packet successfully compare to NST, MRP and MTPC.

Tables 4 and 5 also shows that the performance (PDR and TO) of IDRS is both statistically and scientifically significant different (in bold) from the other three protocols. Hence, the combination of MRP and RDA to classify the radio signal noise pattern has not only

Table 4 p values of the Wilcoxon rank sum test to determine statistical significance of the performance between the routing protocols for the PDR and TO (*Bold highlights* significance value $p < 0.05$)

Protocols	NST:MRP	MRP:MTPC	MRP:IDRS	MPTC:IDRS
	Packet Delivery Rate (%)			
Normal	**3.2011e-06**	0.69072	1.00000	0.95004
Class I	**8.3800e-05**	**0.00029**	**0.02595**	**0.00071**
Class II	**3.2994e-06**	**8.5659e-05**	**0.00102**	0.80234
Class III	**3.5431e-05**	**3.2581e-06**	**3.2994e-06**	**0.02223**
	Transmission Overhead			
Normal	**0.001941**	0.129261	**0.005451**	**4.8774e-06**
Class I	**2.9279e-06**	**2.8064e-05**	**2.9279e-06**	**5.7810e-06**
Class II	**3.3495e-06**	**8.5503e-05**	**3.1688e-06**	**9.1032e-05**
Class III	**2.9430e-06**	**3.1688e-06**	**3.0973e-06**	**3.2173e-06**

Table 5 Vargha-Delaney test to determine scientific significance of the performance between the routing protocols for PDR and TO. The a-values are computed (*Bold highlights* significance value)

Protocols	NST:MRP	MRP:MTPC	MRP:IDRS	MPTC:IDRS
	Packet Delivery Rate (%)			
Normal	**0.00000**	0.54444	0.50000	0.49111
Class I	0.07778	**0.11111**	**0.26000**	**0.86444**
Class II	**0.00000**	**0.07778**	**0.14667**	0.47111
Class III	0.05778	**0.00000**	**0.00000**	**0.74667**
	Transmission Overhead			
Normal	**0.83333**	0.66222	**0.79778**	**0.98222**
Class I	**1.00000**	**0.94889**	**1.00000**	**0.98667**
Class II	**1.00000**	**0.92222**	**1.00000**	**0.92000**
Class III	**1.00000**	**1.00000**	**1.00000**	**1.00000**

allowed the system to response accurately with minimal transmission overhead, but has also maintained a higher PDR.

6.2.3 Trace-based simulation

In order to test the scalability of the IDRS, the IDRS is implemented and tested in NS-2.34 simulation[1] [27]. A trace-based simulation is proposed to compare the IDRS against AODV, NST and MRP in simulation. A trace-based simulation is applied to perform a realistic evaluation using traces database collected from various deployment environments [24]. In this work, we use traces collected from real hardware mote. MTPC is not evaluated because NS-2.34 does not support dynamic power control.

Failure model To simulate the IDRS in a NS-2.34 simulator, the interferences experienced in the real networks need to be implemented in NS-2.34 using a trace-based simulation approach to create the failure in the simulation. In a trace-based simulation, the RSSI traces

[1] Downloadable at http://rtslab.wikispaces.com/file/view/idrs.tar.

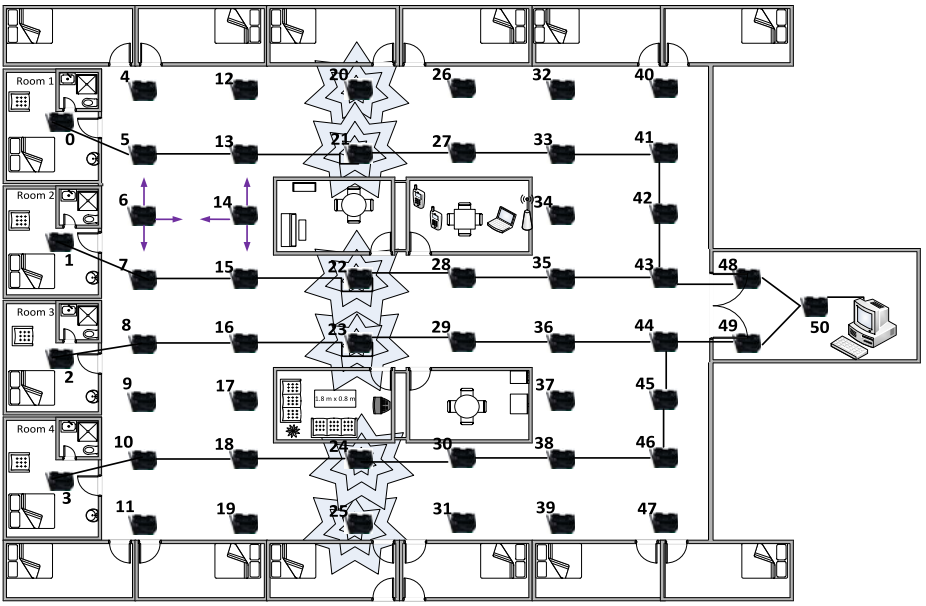
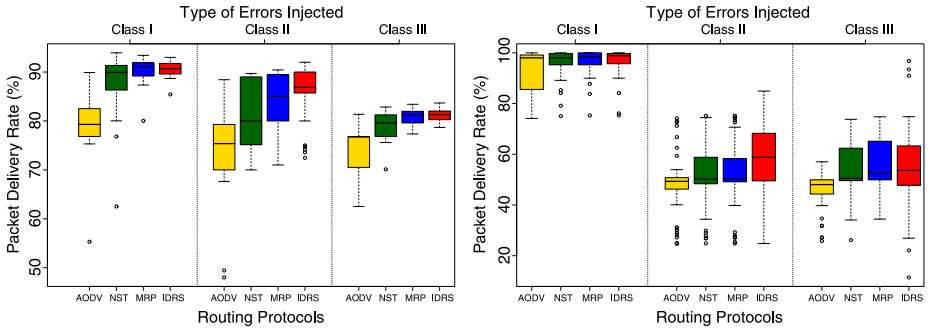


Fig. 14 Network topology based on the indoor deployment for critical health monitoring networks

captured in the hardware experiment are preprocessed and transformed into a series of *on* and *off* patterns according to the values of RSSI (above −87 dB is *off*, otherwise *on*). The ON and OFF patterns have been used in software-based traffic generators to generate WLAN traffics [3]. These *on* and *off* patterns will be used by the node to produce the failure.

50 different failures patterns for each class, each stored in a file are generated. Each of these files will be randomly selected by NS-2.34 to generate the failures observed by each node. In order for RDA to perform the diagnosis during failure, it is necessary to generate 100 sets of raw RSSI pattern for each interference class and store them separately in a file. These RSSI traces are randomly selected by the simulator to run RDA when a failure is detected by the MRP.

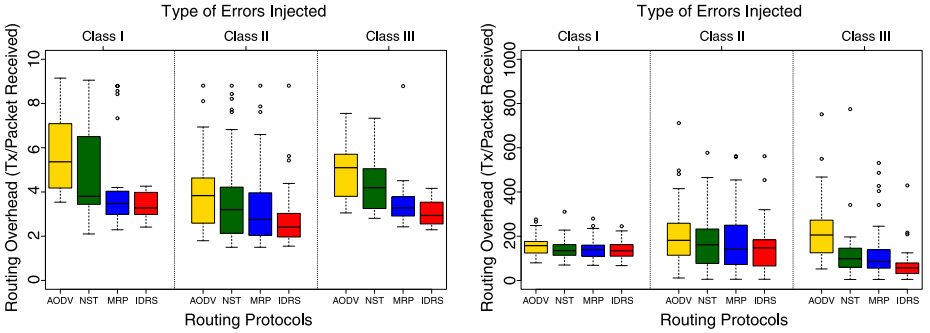
Network setup Two sets of networks are deployed. The first one is a small scale network based on the controlled hardware deployment of 6 nodes that mirrors the hardware experiment and its configurations defined in Sect. 6.2.2. The results from this experiment will be analysed using SPET to measure the *reality gap* achieved by trace-based simulation. In the second experiment, we employ a flat-based topology where each node is usually placed within the transmission range of the neighbouring nodes. The 51 static nodes are deployed across the simulated environment and positioned at the top of the wall as shown in Fig. 14. The nodes are placed 10 m apart with the transmission range set to 14 m to avoid interfering with distant nodes. Packets can only be transmitted to the top, bottom, left or right node within the transmission range of the forwarding node as shown by nodes 6 and 14, but not diagonally. The network is designed with redundant links along the corridor where nodes are place in parallel to each other to form two possible paths between the source and the destination [6]. This alternative node allows individual sensor node to send the packet via the alternative path when the next hop node failed. In a traditional network, it is a common to provide an additional backup link next to the current one to provide redundancy. With this



(a) Simulated PDR for controlled environment

(b) Simulated PDR for indoor deployment

Fig. 15 The Box Whisker plot showing the packet delivery ratio achieved by the AODV, NST, MRP and IDRS for both small controlled (**a**) and large indoor (**b**) environments induced with the 3 different interference classes



(a) Simulated routing overhead for controlled environment

(b) Simulated routing overhead for indoor deployment

Fig. 16 The Box Whisker plot showing the routing overhead produced by the AODV, NST, MRP and IDRS for both small controlled (**a**) and large indoor (**b**) environments induced with 3 different interference classes

additional link, routes can be established quickly without traversing back to the direction of the source and can reduce the network recovery time. To generate the failure for the second experiment, each node 20, 21, 22, 23, 24, and 25 will turn itself on and off according to the on-off pattern observed from the randomly selected failure file. Node 0, 1, 2 and 3 will transmit periodically every 0.2 s. As it is faster and easier to run test the routing protocol in simulation, each simulation is repeated 50 times.

Simulation result Figures 15 and 16 show the performance of the routing protocol induced with different interferences. The overall results have shown that the performance of the IDRS is not affected by the size of the network as shown by the PDR of the small controlled network and large indoor network. However, the communication overhead generated by IDRS is significantly less than MRP, NST and AODV in both small and large networks when induced with a Class III interference compared to Class I and II interferences (Medium A-value > 0.67). The boxplot also shows different trends between the hardware and simulations.

Table 6 Significance test for AODV, NST and MRP against IDRS for the performance metrics PDR, ENG, RT and DLY in both small controlled and large indoor deployments (**Bold** indicates 95 % significance with p-value < 0.05 test and A-value > 0.73 or < 0.27 indicate large effect size)

Interference	Protocols	PDR		RT	
Class	Controlled	p-value	A-value	p-value	A-value
3	AODV/IDRS	**1.16E-014**	**0.01677**	**1.60E-010**	**0.90070**
	NST/IDRS	**0.00011**	**0.26647**	**6.93E-009**	**0.84926**
	MRP/IDRS	0.4426	0.45036	**0.01303**	*0.66000*
2	AODV/IDRS	**1.03E-010**	**0.08424**	**0.00022**	**0.73913**
	NST/IDRS	**0.00046**	*0.30568*	**0.01533**	0.65262
	MRP/IDRS	**0.03661**	0.39933	0.09556	0.62371
1	AODV/IDRS	**2.25E-012**	**0.01435**	**6.14E-010**	**0.92823**
	NST/IDRS	**0.02333**	0.3561	**0.00046**	*0.72228*
	MRP/IDRS	0.80845	0.48421	0.4185	0.55205
Class	Indoor	p-value	A-value	p-value	A-value
3	AODV/IDRS	**0.00020**	**0.25466**	**5.08E-011**1	**0.90387**
	NST/IDRS	0.69946	0.47339	**0.00795**	*0.67923*
	MRP/IDRS	0.78175	0.51774	**0.00641**	*0.67073*
2	AODV/IDRS	**0.00110**	*0.30121*	**0.02827**	0.63333
	NST/IDRS	0.06638	0.38741	0.28315	0.56593
	MRP/IDRS	0.1074	0.40123	0.51853	0.53975
1	AODV/IDRS	0.12441	0.40903	**0.03204**	0.62723
	NST/IDRS	0.93739	0.49504	0.90832	0.5182
	MRP/IDRS	0.97907	0.49823	0.917	0.50643

In the small controlled environment, the PDR of both MRP and IDRS is significantly higher than NST and AODV (above 90 %) in Fig. 15a. Although there are no differences between the PDR for IDRS and MRP, significantly lower routing overhead is observed for IDRS in Fig. 16a for Class III interference with the p-value ≤ 0.01 in Table 6 with scientifically significance a-values. Each time the RDA detects the class III interference, the MRP is alerted. As a result, local repair is avoided; the amount of routing overhead is lowered.

In the large indoor environment, more packets have been delivered by IDRS than AODV in class II and III interference with p-value < 0.001 and a-value < 0.3. Similar to the control environment, the routing overhead generated during class III interference is also significantly lower in IDRS with p-value < 0.006 and a-value < 0.67. Hence, the results from the simulations have shown that by using the RDA to identify the type of interferences in the environment, the reliability and availability of the MRP can be improved significantly even when the network size is increased from 6 to 51 nodes. As a result, the IDRS is scalable.

6.2.4 Computation and memory footprint

In term of the computation and memory footprint introduced by IDRS, MRP, NST and AODV, we capture the processor instruction clock cycle required by the routing protocols to forward 50 packets using the Contiki's Cooja Simulator. Using the Tiny-OS binary compiled

Table 7 The memory footprint (in Byte) required for MRP and IDRS in a TelosB mote and the processor computational (Number of cycles) collected from a Contiki's Cooja Simulations

Overhead	IDRS	MRP	NST	AODV
RAM (KBytes)	5.1	4.0	4.00	3.99
ROM (KBytes)	34.3	32.2	32.0	3.16
Processor (Cycles)	61.7 M	58.1 M	59.2 M	54.7 M

for the hardware, the processor cycle information processed by node 7 in Fig. 11 is computed in the simulator. The statistics collected from 20 runs are given in Table 7 together with the memory footprint obtained during compilation.

From Table 7, the IDRS has the highest mean processor cycles than the other routing protocols required for the execution of the RDA. The mean processing cycle of IDRS is 4.28 % higher than MRP. However, the differences between them are not significant (p-value $=$ 0.06, A-value $=$ 0.30). There are also no differences in the processor to process 50 packets between single mode (NST) and multi-mode (MRP) (p-value $=$ 0.21, A-value $=$ 0.38). The MRP utilises higher (5.5 %) processor cycle than AODV (p-value $=$ 0.03, A-value $=$ 0.30). However, Raghunathan et al. [32] highlighted that the energy consumption required for processing is known to be 70 % less than communication. As a result, the increased in the processor footprint in the multimode routing is acceptable as the number of communications required for routing is reduced significantly, indirectly reduces the energy consumption in the nodes. The code and storage required for the implementation of MRP and IDRS did not increase the memory footprint significantly and can fit in the existing mote.

6.3 Discussion

Although the results from both the simulation and hardware experiments are not the same, both have shown that the proposed IDRS has improved the PDR and reduced the number of transmissions. With the ability to determine the type of interference using the RDA, the appropriate response can be taken to rectify the failure. Both hardware and software results have shown that the PDR of the IDRS is significantly better than AODV, NST and MRP and is scalable. However, the plots in Figs. 12 (PDR for Hardware) and 15a (PDR for Simulation) show that the shape of the graph is not the same. Although the simulation has been designed based on the hardware scenario and the failures introduced are generated using the RSSI patterns collected from the motes, the tests between hardware and software from SPET have shown the results are not the same (Kolmogorov-Smirnov test $p \ll 0.01$). Hence, the trace-based simulation has not failed to produce a realistic results similar to the hardware. Further investigation is required to analyse the cause of these discrepancies.

7 Conclusion

In this paper, an adaptive fault-tolerant routing is presented and evaluated. Using the self-healing properties of the immune system, an Immune-inspired Detection and Recovery System (IDRS) is proposed to detect, identify and recovery from radio failures. The proposed system uses a multi-layer approach that consists of the MPR detection module, the RDA Diagnostic Module, and the Radio Interference Response Module. In the RDM, we have

extended the RDA to diagnose different types of interferences. Our experimental results have demonstrated that RDA can effectively classify the interference based on the RSSI values. By integrating the RDA with the MRP, the accurate response to rectify the network anomalies can be taken and adapted to the observed network conditions. The results in hardware sensor motes show that the IDRS can improve the PDR with Class I and III interference without generating excessive communication overhead. Each node can self-detect, identify and rectify the anomalies that affecting the mote' s radio autonomously. As the signature of normal RSSI can be easily regenerated as required, the IDRS can be made to adapt to the node's current normal environment when the network changes. We also proposed a trace-based simulation to evaluate the scalability of IDRS. The results from the simulation have shown that the IDRS is scalable and similar performance improvements observed in hardware have been produced. However, results obtained from simulation do not exhibit the same distribution as the hardware as demonstrated by SPET especially for class I interference requiring a further investigation on the ability of the trace-based simulation to represent the real world.

References

1. Balakrishnan H, Padmanabhan V, Seshan S, Katz R (1997) A comparison of mechanisms for improving tcp performance over wireless links. IEEE/ACM Trans Netw 5(6):756–769
2. Boers NM, Nikolaidis I, Gburzynski P (2010) Patterns in the RSSI traces from an indoor urban environment. In: Proceedings of international workshop on computer aided modeling, analysis and design of communication links and networks. IEEE Press, New York, pp 61–65
3. Botta A, Dainotti A, Pescapé A (2010) Do you trust your software-based traffic generator? IEEE Commun Mag 48(9):158–165
4. Candea G, Cutler J, Fox A (2004) Improving availability with recursive microreboots: a soft-state system case study. Perform Eval 56(1):213–248
5. Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surv 41(3):1–58
6. Chipara O, Lu C, Bailey T, Roman G (2010) Reliable clinical monitoring using wireless sensor networks: experiences in a step-down hospital unit. In: Proceedings of the 8th conference on embedded networked sensor systems. ACM, New York, pp 155–168
7. Davoudani D, Hart E, Paechter B (2007) An immune-inspired approach to speckled computing. In: Castro L, Zuben F, Knidel H (eds) Artificial immune systems. Lecture notes in computer science, vol 4628. Springer, Berlin Heidelberg, pp 288–299
8. Gnawali O, Fonseca R, Jamieson K, Moss D, Levis P (2009) Collection tree protocol. In: Proceedings of the 7th conference on embedded networked sensor systems. ACM, New York, pp 1–14
9. Gomez C, Salvatella P, Alonso O, Paradells J (2006) Adapting AODV for IEEE 802.15.4 mesh sensor networks: theoretical discussion and performance evaluation in a real environment. In: Proceedings of the international symposium on world of wireless, mobile and multimedia networks. IEEE Press, New York, pp 159–170
10. Gutierrez J, Naeve M, Callaway E, Bourgeois M, Mitter V, Heile B (2001) IEEE 802.15.4: a developing standard for low-power low-cost wireless personal area networks. Networks 15(5):12–19
11. Hart E, Timmis J (2008) Application areas of AIS: the past, the present and the future. Appl Softw Comput 8(1):191–201
12. Hilder J, Owens N, Neal M, Hickey P, Cairns S, Kilgour D, Timmis J, Tyrrell A (2012) Chemical detection using the receptor density algorithm. IEEE Trans Syst Man Cybern, Part C, Appl Rev 42(6):1730–1741
13. Hsu L, King C, Banerjee A (2007) On broadcasting in wireless sensor networks with irregular and dynamic radio coverage. In: International conference on parallel processing. IEEE Press, New York, pp 55
14. IEEE: Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf (2006) [Online; accessed 1-March-2013]
15. Karlof C, Wagner D (2003) Secure routing in wireless sensor networks: attacks and countermeasures. In: Proceedings of the 1st international workshop on sensor network protocols and applications. IEEE Press, New York, pp 113–127

16. Ko J, Terzis A (2010) Power control for mobile sensor networks: an experimental approach. In: Proceedings of the 7th annual communications society conference on sensor mesh and ad hoc communications and networks. IEEE Press, New York
17. Lau H, Bate I, Cairns P, Timmis J (2011) Adaptive data-driven error detection in swarm robotics with statistical classifiers. Robot Auton Syst 59(12):1021–1035
18. Levis P, Madden S, Polastre J, Szewczyk R, Whitehouse K, Woo A, Gay D, Hill J, Welsh M, Brewer E, Culler D (2005) TinyOS: an operating system for sensor networks ambient intelligence. In: Weber W, Rabaey JM, Aarts E (eds) Ambient intelligence, Chap. 7. Springer, Berlin Heidelberg, pp 115–148
19. Lim TH, Bate I, Timmis J (2011) Multi-modal routing to tolerate failures. In: Proceedings of the 7th international conference on intelligent sensors, sensor networks and information processing. IEEE Press, New York, pp 211–216
20. Lim TH, Bate I, Timmis J (2012) Validation of performance data using experimental verification process in wireless sensor network. In: Proceedings of the 16th conference on emerging technologies factory automation. IEEE Press, New York
21. Lin S, Zhang J, Zhou G, Gu L, Stankovic J, He T (2006) ATPC: adaptive transmission power control for wireless sensor networks. In: The 4th international conference on embedded networked sensor systems, pp 223–236
22. Lin S, Zhou G, Whitehouse K, Wu Y, Stankovic J, He T (2009) Towards stable network performance in wireless sensor networks. In: Proceedings of the 30th real-time systems symposium. IEEE Press, New York, pp 227–237
23. Liu H, Li J, Xie Z, Lin S, Whitehouse K, Stankovic JA, Siu D (2010) Automatic and robust breadcrumb system deployment for indoor firefighter applications. In: Proceedings of the 8th international conference on mobile systems, applications, and services. IEEE Press, New York, pp 21–34
24. Marchiori A, Guo L, Thomas J, Han Q (2010) Realistic performance analysis of wsn protocols through trace based simulation. In: Proceedings of the 7th ACM workshop on performance evaluation of wireless ad hoc, sensor, and ubiquitous networks. ACM, New York, pp 87–94
25. Murphy K, Travers P, Walport M (2012) Janeway's immunobiology, vol 7. Garland Science, New York
26. Ngai E, Liu J, Lyu M (2006) On the intruder detection for sinkhole attack in wireless sensor networks. In: Proceedings of the international conference on communications, vol 8. IEEE Press, New York, pp 3383–3389
27. NS2: The network simulator ns-2 (2002). http://www.isi.edu/nsnam/ns/ [Online; accessed 1-February-2013]
28. Ong K, Yue S, Ling K (2010) Implementation of fast Fourier transform on body sensor networks. In: Proceeding of the international conference on body sensor networks. IEEE Press, New York, pp 197–202
29. Owens N, Greensted A, Timmis J, Tyrrell A (2012) The receptor density algorithm. Theoretical Computer Science
30. Perkins C, Royer E (1999) Ad-hoc on-demand distance vector routing. In: Proceeding of the 2nd workshop on mobile computing systems and applications. IEEE Press, New York, pp 90–100
31. Polastre J, Szewczyk R, Culler D (2005) Telos: enabling ultra-low power wireless research. In: Proceeding of the 4th international symposium on information processing in sensor networks. IEEE Press, New York, pp 364–369
32. Raghunathan V, Schurgers C, Park S, Srivastava M (2002) Energy-aware wireless microsensor networks. IEEE Signal Process Mag 19(2):40–50
33. Schaust S, Szczerbicka H (2011) Applying antigen-receptor degeneracy behavior for misbehavior response selection in wireless sensor networks. In: Proceedings of the 10th international conference on artificial immune systems. Springer, Berlin Heidelberg, pp 212–225
34. Srinivasan K, Dutta P, Tavakoli A, Levis P (2010) An empirical study of low-power wireless. ACM Trans Sens Netw 6(2):1–49
35. Szewczyk R, Polastre J, Mainwaring A, Culler D (2004) Lessons from a sensor network expedition. In: Karl H, Wolisz A, Willig A (eds) Wireless sensor networks. Lecture notes in computer science, vol 2920. Springer, Berlin Heidelberg, pp 307–322
36. Trinidad M, Valle M (2009) Reliable event detectors for constrained resources wireless sensor node hardware. EURASIP J Embed Syst 2009:7
37. Vargha A, Delaney H (2000) A critique and improvement of the CL common language effect size statistics of McGraw and Wong. J Educ Behav Stat 25(2):101–132
38. Wallenta C, Kim J, Bentley P, Hailes S (2010) Detecting interest cache poisoning in sensor networks using an artificial immune algorithm. Appl Intell 32(1):1–26
39. Wang P, Akyildiz I (2011) Spatial correlation and mobility-aware traffic modeling for wireless sensor networks. IEEE/ACM Trans Netw 19(6):1860–1873

40. Wilcoxon F (1945) Individual comparisons by ranking methods. Biom Bull 1(6):80–83
41. Zacharias S, Newe T, O'Keeffe S, Lewis E (2012) Identifying sources of interference in rssi traces of a single IEEE 802.15.4 channel. In: Proceeding of the 8th international conference on wireless and mobile communications, pp 408–414
42. Zou Y, Chakrabarty K (2007) Redundancy analysis and a distributed self-organization protocol for fault-tolerant wireless sensor networks. Int J Distrib Sens Netw 3(3):243–272