Chapter 25
HCDSR: A Hierarchical Clustered Fault Tolerant Routing Technique for IoT-Based Smart Societies

Thaha Muhammed, Rashid Mehmood, Aiiad Albeshri, and Ahmed Alzahrani

25.1 Introduction

Urban population has increased vastly in the recent years. The United Nations Human Settlements Program (UN-Habitat) [41] has foreseen it to be 10 billion by 2050, which is two-thirds of the current population on earth. The cities will have to deal with pressing issues such as public safety, efficient transportation, energy consumption, environmental sustainability, and expense reduction. These pressing issues have led to smart city paradigm which aims to plan and develop efficient urban cities in future.

The past decade has witnessed the advancement of Internet of Things (IoT) [37, 40] especially the sensing technology [25]. In addition to the widespread development of sensors, improvement in big data computing infrastructure has enabled the collection of huge amount of heterogeneous data produced daily by urban spaces [11]. Urban spaces produce data related to temperature, weather, pollution, traffic control, the mobility of people, and resource consumption (water and electricity) which can be analyzed to improve the services provided and make the environment greener. Smart cities rely on sensors, webcams, IoT systems, wireless sensor networks, databases, ubiquitous devices, and many other frameworks that collect, process, and take informed decisions based on the data [5]. A survey on data fusion and IoT for smart ubiquitous environments can be seen in [6].

T. Muhammed (✉) · A. Albeshri · A. Alzahrani
Department of Computer Science, FCIT, King Abdulaziz University, Jeddah, Saudi Arabia
e-mail: m.thaha.h@ieee.org; aaalbeshri@kau.edu.sa; asalzahrani@kau.edu.sa

R. Mehmood
High Performance Computing Center, King Abdulaziz University, Jeddah, Saudi Arabia
e-mail: RMehmood@kau.edu.sa

R. Mehmood et al. (eds.), *Smart Infrastructure and Applications*,
EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-13705-2_25

Wireless sensor networks (WSNs) are one of the atomic components of IoT. Data acquisition for IoT applications requires wireless sensor network and is the link between the real world and the digital world. WSNs play a major role in building interconnected urban territories and are critical to smart cities. WSNs consist of small low power sensor nodes that can sense, process, and wirelessly communicate with each other. The sensors are devices with limited battery, storage, size, and computational power. The sensors nodes sense data and forward it to a base station known as sink for further processing of data by IoT systems. Intelligent monitoring and management of smart cities are possible through IoT. WSNs are used in a number of time-critical smart city applications such as agriculture monitoring [43], intruder detection [19], disaster management, health care, mobile object tracking, environment monitoring [19], intelligent transport system (obstacle detection, collision warnings and avoidance, traffic monitoring) [16, 42, 47], vehicular ad-hoc networks [35], energy monitoring in smart grids [38], and home/office automation systems (HOS) [20].

Since WSNs are deployed in harsh and hostile conditions they are susceptible to frequent errors. The occurrence of faults results in disruption of the network or worse in the failure of the network. This might lead to human, economic, environmental loss since the sensors are used in many safety critical applications. Another source of a fault in WSN is the power [25]. Since the WSNs work unattended in a hostile environment it is not feasible to replenish the batteries of the sensors. Moreover, various hazard might cause the power to run out, which results in a node failure. Data transmission consumes a major portion of energy [46]. Hence prolonging energy in WSN becomes a critical and challenging issue [1, 3, 9, 15, 32]. A detailed discussion on possible faults in wireless sensor networks has been discussed in [39].

It is required that the data collected by the sensors on critical events should not be of low quality [18, 21, 22] that might lead to important information loss, but often random link failures occur that disrupt communication in the network. All these issues point to the necessity of fault tolerance techniques that would provide techniques to mask these faults and provide the expected services, in the presence of faults. Major disadvantages of existing techniques are a high dissipation of energy, large mean time to repair (MTTR), and the use of extra software and hardware [44, 45].

Clustering has been used by the researchers to reduce the energy consumption in WSN [4]. In a clustered WSN, the sensors are clustered into mutually exclusive clusters. A sensor node is associated with a single cluster and each cluster has a cluster head (CH) that aggregates the data from the nodes associated with it and transmits it to the base station (sink). The transmission of a large amount of data by the CHs leads to depletion of energy and consequently leads to the death of the CH. To prevent this many researchers [28–30] have proposed using special gateways with higher initial energy, but the use of special gateways is not feasible when the network is deployed randomly in inaccessible locations. This also creates a problem with clustering and reclustering when randomly deployed. Consequently, many researchers have used techniques such as multipath routing, backbone scheduling, and node scheduling. Researchers have also proposed various clustering techniques to improve ad-hoc network performance, see, e.g., [7, 8, 33, 34, 36].

In this paper, we propose a new fault tolerant routing algorithm based on modified dynamic source routing (DSR) on a clustered, hierarchical sensor network for IoT applications. We use a vice cluster head that takes over the duties of the CH on the failure of a CH. Moreover, we use multiple paths that have been prioritized and sorted on the basis of a cost function that takes into consideration the total energy in a path and the distance from the source to sink. Furthermore, we use energy thresholds to decide the CHs that would participate in the routing process. One of the major advantages of the technique is that the mean time to repair (MTTR) for this technique is small. We simulate our algorithm and compare our algorithm with DFTR [12], a distributed fault tolerant algorithm and LEACH (low-energy adaptive clustering hierarchy) [25], a well-known routing algorithm. Metrics such as the number of alive nodes, total energy consumption of the network, and total packets transmitted to the sink are compared measured for all the three techniques. Based on these metrics it was observed that HCDSR performs better than the other techniques. This paper extends our earlier work [40]. The contributions to this work can be summarized as follows:

- We present a survey of fault tolerant and energy-efficient routing techniques for WSNs.
- We propose a new up-to-date taxonomy for fault tolerant strategies for WSNs.
- We provide a brief qualitative analysis and comparison of latest fault tolerant strategies for WSNs.
- We propose a new energy-efficient fault tolerant routing strategy called heterogeneous modified dynamic source routing (HCDSR).
- We simulate the proposed technique.
- The results from the proposed technique are compared with two current techniques, LEACH [25] and DFTR [12], demonstrating better performance.

The rest of the article is organized as follows. Section 25.2 discusses the proposed taxonomy for fault tolerant techniques in WSN. In Sect. 25.3, we discuss the state-of-the-art fault tolerant techniques for WSN. We also do a qualitative analysis of FT techniques in WSN. In Sect. 25.4, we discuss the system model and Sect. 25.5 introduces our proposed FT routing technique. Section 25.6 presents the simulation of the proposed technique. It also presents the comparison with techniques to validate our proposed technique. Section 25.7 concludes the paper.

25.2 Taxonomy

Fault tolerance techniques in wireless sensor networks can be classified according to two criteria, namely based on the phase at which the fault tolerant technique triggers and based on the origin of faults in WSN. Based on these criteria fault detection

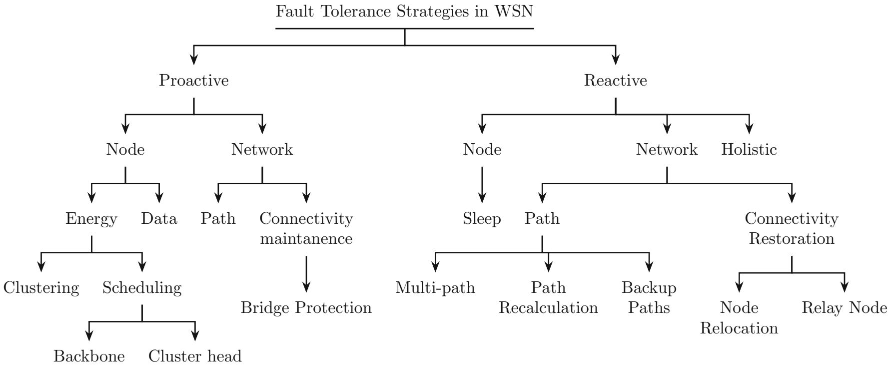


Fig. 25.1 Proposed taxonomy of fault tolerant techniques in WSN

techniques in WSN can be classified as (1) proactive and (2) reactive as shown in Fig. 25.1.

25.2.1 Proactive Techniques

Proactive techniques in WSN proactively and sensibly use the existing resources of the wireless sensor to extend the lifetime of the network or prevent the fault from occurring. These techniques take preemptory action against potential faults. Based on the origin of faults these techniques can be classified (1) node-based techniques, (2) network-based techniques, and (3) holistic techniques.

Node-Based Techniques The node-based proactive techniques can be further classified as (1) energy-based fault tolerance technique and (2) data fault tolerance. Energy-based fault tolerance increases the mean time to failure and the lifetime of the network. This strategy uses techniques such as clustering of sensor nodes, hibernation of nodes, and scheduling nodes and backbone of the WSN. Proactive data fault tolerant techniques help in recovering from data faults. One of the major techniques of data fault tolerance is the dual transmission of the same value and comparison of these data to detect the faults.

Network-Based Techniques It comprises of mainly two techniques, namely (1) connectivity maintenance technique and (2) multipath routing. Connectivity maintenance techniques increase the lifetime of network using various algorithms. Bridge protection algorithm is an example of connectivity maintenance algorithm that increases the lifetime of WSNs comprising bridged nodes. Data is sent through multiple paths to increase the redundancy and tolerate network fault in multipath techniques.

25.2.2 Reactive Techniques

Reactive techniques trigger the fault tolerant strategy on the occurrence of the faults. This strategy waits for the faults to occur and then adjusts or reacts to the fault by starting the recovery process. These techniques can also be further classified based on the origin of the faults as (1) node-based, (2) network-based, and (3) holistic-based technique.

Node-Based Techniques Node-based reactive techniques are used to recover from node failures. It consists of strategies like switching to the sleeping backup node on the occurrence of node failure.

Network-Based Faults Network-based reactive techniques consist of using multiple paths, backup paths, and path recalculation in case of network/link failure. Moreover, for restoring the connectivity, extra nodes are deployed or the existing nodes are repositioned.

Holistic Techniques These are the techniques that can deal and recover from both network- and node-based faults. They provide a complete fault tolerance for various faults.

25.3 Previous Work

In this section we shall discuss the existing work related to WSN fault tolerance techniques. In this section we shall discuss the current work related to WSN fault tolerance techniques. Tables 25.1 and 25.2, respectively, give the advantages and disadvantages of various fault tolerant techniques for WSNs that shall be discussed in this section. Tables 25.3 and 25.4 give a comparison of some of the FT routing techniques that shall be discussed ahead.

25.3.1 Proactive Techniques

Zhao et al. [49] propose a sleep scheduling technique, called virtual backbone scheduling (VBS). In this technique, we form multiple backbones that overlap with each other. Data is transmitted to the sink using only these backbones. The nodes that are not part of the backbone do not participate in transmission to save energy. The energy consumption of the nodes is balanced by rotating the backbones. This results in a longer lifetime of the network. Selection of the backbones that increases the network lifetime is an NP-hard problem and hence the authors propose three techniques to solve this. These schemes are based on (1) schedule transition graph (STG), (2) virtual scheduling graph (VSG), and (3) iterative local replacement (ILR). The longest path in a schedule transition graph corresponds to

Table 25.1 Comparison of advantages of various fault tolerant techniques in WSN

Protocol	Taxonomy	Advantages
VBS [49]	Proactive	– Energy-efficient routing – Increases lifetime of network
BPA [27]	Proactive	– Reduction in message overhead – Increases energy efficiency – Balance the energy in the network
PASC_AR [17]	Proactive	– Increases network lifetime
DFTR [12]	Proactive	– Reduces energy consumption – Increases gateway lifetime
MDSR [2]	Reactive	– Energy-efficient routing – Fault tolerant routing – Total throughput increases due to higher lifetime of the network
B^{3}FT [23]	Reactive	– Energy efficient – Energy balanced – Cluster head tolerance – Increases the time span of network (CH + nodes)
FT PSO [13]	Reactive	– Increases lifetime of network – Energy balanced
FTEAM [26]	Reactive	– Increases lifetime of cluster heads – Improves reliability
IFTF [24]	Holistic	– Holistic approach detects permanent node failures – Monitors network quality – Determines source and cause of the fault

Table 25.2 Comparison of disadvantages of various fault tolerant techniques in WSN

Protocol	Disadvantages
VBS [49]	– Node failures might require recalculating the backbones as nodes overlap among backbones
BPA [27]	– Trade-off between time and residual energy
PASC_AR [17]	– Reduction of sensing accuracy due to the sleep mode of nodes
DFTR [12]	– Fixed gateways not always suitable or plausible – Clustering difficult when there is fixed gateways
MDSR [2]	– Reduction in throughput of network for any given time period as all nodes not involved in transmission
B^{3}FT [23]	– Use of extra hardware as gateways
FT PSO [13]	– Does not handle fault tolerance if no gateway in range
FTEAM [26]	– Trade-off between accuracy and lifetime – Only reliable when rate of change of sensed value is very small inside the cluster
IFTF [24]	– Four percent increase in message overhead – Does not consider the computation overhead – Does not provide a specific technique for recovering from fault

Table 25.3 Comparison of fault tolerant routing techniques

Protocol	Energy savings	Phases	Message reduction	FT technique
MDSR [2]	✓	1	X	Multiple routes stored
E3BFT [23]	✓	3	✓	Rejoin new cluster head
FT PSO [13]	✓	3	✓	Selection of alternate gateways
DFTR [12]	✓	3	✓	Selection of alternate gateways

Table 25.4 Comparison of clustering techniques used in selected fault tolerant routing schemes

Protocol	Clustering	Clustering type	Intra cluster routing	Clustering technique
E3BFT [23]	Yes	Proactive	Multi-hop	Using residual energy, routing overhead, and node-gateway distance
MDSR [2]	X	NA	NA	NA
FT PSO [13]	Yes	Proactive	Multi-hop	Using special gateways
DFTR [12]	Yes	Reactive	Multi-hop	Using special gateways

the backbone that provides maximum network lifetime. STG models structure and energy separately. In VSG, multiple virtual nodes are created from sensor nodes in such a way that the energy of corresponding sensor nodes is represented by their degrees. STG and VSG are both centralized techniques. ILR is a distributed technique that uses local information for switching backbone. The switching is done node by node iteratively and each node decides the next node to be switched by analyzing the energy of neighboring nodes.

Khan et al. [27] propose a fault tolerant algorithm for bridge protection in WSNs. The fragmentation of WSN due to the various events might result in the formation of a bridge node, which maintains the network connectivity. Bridge nodes are nodes whose removal results in communication failure between the fragments of WSN. The authors propose a bridge protection algorithm with manifold goals. Primarily to prevent the bridge node(s) from prematurely exhausting the energy and secondarily for preventing the formation of new bridge nodes from its neighboring nodes and to maintain the minimal functionality of the network with minimal interference. The authors modify the functionalities at the bridge node, gate nodes (non-bridge neighboring nodes of bridge node(s) on the side without sink), and fan-out nodes (non-bridge neighboring nodes of bridge node(s) on the side with sink). Most message reasoning is shifted from sink to gate nodes. Heartbeat messages will no longer be sent by the nodes, instead the gate nodes will process and detect if a node fails and send a message only on failure to save energy. Occlusion reasoning is used to avoid obsolete and redundant messages being sent to sink to save energy. To avoid creating load on fan-out nodes, the bridge node splits the traffic based on round-robin scheduling. The fan-out nodes further split the traffic to avoid the traffic coalescing at a single node to result in energy depletion. The routing alternatives chosen by the fan-out and bridge nodes are selected using Delaunay triangulation of the nodes and further reducing it into relative neighboring graphs, which will further diminish power consumption, extending the life of bridge nodes.

Boucetta et al. [17] propose an energy-efficient fault tolerant scheduling algorithm, called power aware scheduling and clustering protocol with adaptive redeployment (PASC_AR). In this technique, all nodes are considered to have the initial same energy level and same capabilities. The network is partitioned into zones. All sensor nodes which sense identical values are considered to be in the same zone. The network is clustered geographically based on node location. Only one node from each node will be active and this node will be assigned as the cluster head. The cluster head creates a TDMA schedule that is used to select a new cluster head. The cluster head is selected in rotation from the nodes in the zone based on this schedule. The rest of the nodes are put to sleep. The nodes at the sink tend to fail earlier due to higher traffic. To recover from this, the authors propose a cascading movement of the redundant sensor nodes toward zones near the sink. PASC_AR maintains network connectivity and coverage by preventing routing holes.

Azharuddin et al. [12] propose an energy saving and FT routing technique, called DFTR that not only deals with energy utilization of cluster heads but also their fault tolerance. The routing is done based on the following criteria: (1) gateway to next hop gateway distance, (2) next-hop gateway to base station distance, and (3) energy remaining at the next-hop gateway. In the cluster setup phase, the clusters are formed and data is transmitted to the base station by the cluster heads. In this technique, the cluster heads are classified into three categories: (1) forward cluster head (ForwardCH), (2) backward cluster head (BackwardCH), and (3) orphan cluster head (OrphanCH). ForwardCH (G) is a set of neighboring cluster heads of the cluster head G closer to the base station. BackwardCH (G) is a set of neighboring cluster heads of the cluster head G which are farther from base station than G and ForwardCH in different routes. OrphanCHs are cluster heads that are not included in ForwardCH and BackwardCH. If the next-hop faulty cluster head is from ForwardCH, then a new next-hop gateway from ForwardCH is selected on the basis of minimum cost. If the ForwardCH for a gateway is empty, then the new next-hop gateway is selected from BackwardCH on the basis of minimum cost. Moreover, if the gateway doesn't have any ForwardCH and BackwardCH, then that CH is said to be OrphanCH and a gateway that has BackwardCH can help out the OrphanCH.

25.3.2 Reactive Techniques

Rana [2] proposes a modified dynamic source routing (DSR) algorithm offering energy-efficient, fault tolerant routing. The major features of this technique are (1) non-usage of nodes below certain energy threshold in the routing process, and (2) two routes cached between source and destination. Initially in the route discovery phase, the source node floods the network with route request packets (RREQ). It appends to this packet, its energy level at time t. Only the nodes with energy level above a threshold energy participate in the flooding process. Each node that participates in this process appends its energy level to the RREQ packet. Multiple packets traveling through multiple paths may reach the destination D. Two best paths are selected based on the highest average energy of the paths. The destination waits for a time period T, for the arrival of all possible packets before making the final decision. Once the decision has been made the destination sends a route reply packet (RREP) informing the source about the selected paths. In the case of failure in the primary path, the secondary path is immediately used and the neighboring nodes send route error message (RERR) to update the routing information.

Gupta et al. [23] propose an energy-efficient fault tolerant clustering algorithm, named B^3FT. In this technique, the authors discuss fault tolerance for cluster heads without the redundant usage of cluster heads. Initially during the bootstrap process, all the cluster heads broadcast a HELLO message which consists of the cluster head ID, distance from base station, remaining energy, and distance from the base station. Sensor nodes that receive this message are considered to be in the range of the cluster head. All nodes that did not receive the HELLO packet broadcast an REQ message. Nodes which receive the REQ packet reply with RES packet that consists of its ID, distance from the base station, and the overhead of cluster head it is the associated with. In the next phase, the sensors join various cluster heads by considering the routing overhead, the distance between the nodes and the gateways, and the remaining energy of relay nodes. If a sensor node couldn't find a cluster head, then the neighbor node with the higher remaining energy and minimal overhead acts as a cluster head for that node. In the case of cluster head failure, all sensor nodes associated with this node broadcast REQ message and wait for an REP message. If a node receives REP message from cluster heads, then the node will associate itself with cluster head that has minimum cost else it joins the node with minimum cost.

Azharuddin et al. [13] propose a fault tolerant clustering- based routing algorithm based on particle swarm optimization. This routing technique has two stages: (1) network setup stage and (2) steady state. During the network setup phase, the base station assigns ID to all the nodes and gateways. The nodes broadcast these IDs and are assigned to gateways depending on the distance from the nodes. The gateways send this local node information to base station, where the load (number of packets received) of each gateway is calculated. This can be calculated as follows:

$$P_r(g_i) = \begin{cases} \sum\{P_r(g_j) = g_i,\ g_j \in G\} & \text{If } NextHop(g_j) = g_i,\ \forall g_{j \in G} \\ 0 & \text{Otherwise} \end{cases} \tag{25.1}$$

where $P_r(g_i)$ is the number of data packets received by each gateway g_i per round.

Let E_R, $E_T(g_i, NextHop(g_i))$, and $E_{intraclstr}$ be the energy utilization due to receiving of data by the gateways, sending of a data packets to the next-hop gateway, and energy utilization of a gateway g_i due to various intra-cluster activity. The power utilized by a gateway g_i for a round can be calculated as follows:

$$E_{consump}(g_i) = P_r(g_i) \times E_R + (P_r(g_i) + 1) \times E_T(g_i, NextHop(g_i)) + E_{intraclstr}$$

The lifetime of g_i with remaining residual energy $E_r(g_i)$ can be calculated as follows: $Lifetime(g_i) = E_r(gi)/E_{consmp}(g_i)$. We maximize the lifetime of the gateway with minimum lifetime by minimizing the routing load over the gateway. This is achieved with the help of particle swarm optimization. In case of a gateway failure during routing, the preceding gateway to the failed gateway broadcasts a HELP message. The neighboring gateways respond and the gateway with maximum lifetime toward the base station is selected as the new gateway. If no gateways respond, then the gateway is assumed as dead.

Hezaveh et al. [26] propose a technique called fault tolerant and energy aware mechanism (FTEAM). In this technique, we identify overlapped sensor nodes and put the nodes with highest residual energies to sleep so they can be used as a cluster head in case of cluster head failures. FTEAM consists of four phases: (1) cluster formation, (2) error free, (3) cluster failure, and (4) error recovery phase. In the initial phase, clusters are formed as nodes associate themselves with cluster heads depending upon the distance. The cluster heads determine the overlapped nodes with similarly sensed data and put the nodes with higher residual energy to sleep. In the error free state, the nodes send sensed data to cluster heads, which aggregates the data and sends it to the base station. During this state many nodes die and the energy of the cluster heads gradually decline below a certain threshold. This results in switching to error recovery phase wherein the sleeping nodes are awoken and the 5% of nodes with the highest energy is chosen as new cluster heads. When all sleeping nodes die later, the network fails.

25.3.3 Holistic Techniques

Dima et al. [24] propose an integrated fault tolerance framework (IFTF) which holistically considers all the fault issues in WSNs. IFTF monitors the wireless sensor network and can detect and diagnose application level faults, network layer faults, and establish the root cause of the fault. To achieve this IFTF uses two services, (1) a network level diagnosis sub-service for identifying network level faults such as node and link failure and (2) an application testing sub-service for detecting application anomalies. IFTF manager coordinates these two sub-services together to deal with complex fault scenarios. Application testing sub-service is responsible for ascertaining the adeptness of the system in accomplishing its functionalities. It performs functional testing by comparing an input value with expected value. The network diagnosis service monitors the energy levels of the nodes and the connectivity of the nodes to their neighbors. It uses a two-phase detection algorithm for detecting permanently faulty nodes.

25.4 Network and Radio Model

We consider a clustered WSN which consists of a single base station/sink and multiple clusters of sensor nodes. The sensor nodes in each cluster are normal nodes that are responsible for sensing and transmitting the data to their respective cluster heads (CH). All the nodes and the cluster heads are considered to be homogeneous with identical initial energy levels. The CHs are also normal nodes with the same energy constraints as that of sensing nodes. The CHs receive the sensed data, aggregate the data, and forward it to the base station. Direct data transmission occurs if the base station is one hop away from the CH else the aggregated data is forwarded to the CH closer to the base station. The nodes are deployed randomly as in smart dust model. The sensor nodes and CHs are considered immobile. There is only a single base station which is stationary and has an inexhaustible power supply. All sensor nodes have equivalent bi-directional communication range. All the wireless links are assumed to be symmetric so as to compute the distance between the nodes based on the received signal strength [48]. CSMA/CA MAC protocol is used by the CHs for communicating with base station [48]. For energy consumption analysis we only consider the energy used due to transmission and receiving of data since radio is the most power consuming parts as the consumption due to sensing and computing is negligible.

In this technique we use a radio model that is used in [25]. The energy dissipated E_T due to the transmission of a message of size l-bit between two nodes separated by a distance d is given by

$$E_T(l,d) = \begin{cases} l(E_{elec} + \varepsilon_{FS} \times d^2) & for\ (d < d_0) \\ l(E_{elec} + \varepsilon_{MP} \times d^4) & for\ (d > d_0) \end{cases} \tag{25.2}$$

where $d_0 = \sqrt{\varepsilon_{FS}/\varepsilon_{MP}}$, E_{elec} is the electronic energy required by the circuit, ε_{FS} and ε_{MP} are the transmit amplifier parameters that represent the energy required by the amplifier in free space and multipath models, respectively. The energy dissipation at the receiver sensor node for a message of size l-bit is given by

$$E_R(l) = l \times E_{elec} \tag{25.3}$$

Moreover, the energy consumed for fusing $l-$bits can be given by

$$E_F(l) = l \times E_{df} \tag{25.4}$$

where E_{df} is the energy incurred due to fusing of one bit data.

25.5 Proposed Technique

The proposed technique has four phases: (1) setup phase, (2) route determination phase, (3) data communication phase, and (4) fault recovery phase.

25.5.1 Network Setup

Initially the network will be in the setup phase and all the sensor nodes send a HELLO message to the sink. The sink then assigns an ID to all sensor nodes. During the setup phase, we use any of the standard clustering algorithm to cluster the network and assign a cluster head to each cluster. The cluster head in each of the cluster sends a HELLO message to all its nodes with specific power and based on the strength of the signal received, it finds the nearest node to itself. The nearest node to the cluster head in each cluster is assigned as the vice cluster head. The sink then broadcasts a HELLO message using specific amount of power to all the cluster heads. The sink calculates the distances to each sensor node using the radio strength and this distance is sent back to the cluster head. The setup phase ends and the communication phase starts wherein the nodes send their data to the cluster heads and the cluster heads will fuse multiple identical values into a single value [25]. After a certain amount of time the network switches back to the setup phase so as to balance the energy of the nodes in the network. Subsequently, it enters the communication phase and this process continues until the network encounters a fault.

25.5.2 Route Discovery and Routing Algorithm

We develop our routing technique on top of the foregoing medium access control (MAC) layer. The major steps in our routing technique are given below:

Step 1: Initially in the route determination phase, each cluster head broadcasts an REQ packet similar to that of dynamic source routing (DSR). The REQ initially consists of the source ID, destination ID, the energy of the each cluster head, and the distance to the sink that was obtained during the bootstrap process.

Step 2: This REQ packet is flooded among other cluster heads, and each cluster adds to the packet their respective ID, energy level, and the distance to the base station.

Step 3: We define an energy threshold level. Any cluster head that has energy level below this threshold will not participate in the flooding process.

Step 4: The broadcasted REQ packets reach the destination. For each cluster head, the sink starts a timer on the arrival of the first REQ packet from that cluster head. The sink will wait for more packets till timer expires. Once the timer expires the sink will analyze and select the routes for each cluster head based on the remaining energy level and the sum of distance of all cluster heads in the path to the sink. Based on this the routes for each cluster head are prioritized and are given priority numbers P1, P2,. . ., Pn.

Step 5: Thereafter, the sink sends an REP message to the cluster heads through all the discovered routes for the cluster heads. The REP message consists of the ID of the nodes that are in the path and the priority of that path.

Step 6: Once all REP messages reach the cluster heads they save the routes on basis of their priority and the route with priority P1 will be used for sending data to base station. The intermediate nodes between the source and destination also save the routes.

Step 7: Once the routes are selected, the cluster heads pass the route information to their respective vice cluster heads and the vice cluster head resume their sleep state after storing this information.

The routing algorithm has been given in Algorithm 1.

25.5.3 Fault Tolerance

In this technique we only consider faults in routing especially disruption of route due to failure of cluster heads. We can consider the following cases of failures:

Failure of the source cluster head. When the source cluster head fails, the vice cluster head takes over the job of the cluster head. The routing table is already present in the vice cluster head as explained before.

Failure of the intermediate cluster head. When the data from the source cluster head is sent ahead and one of the intermediate cluster heads fails then the failed cluster head sends an error message (ERR) to the preceding cluster head. The preceding cluster head will switch its route from primary to secondary route and the faulty cluster head will be replaced by the vice cluster head. If the secondary route also fails, then it will use the tertiary route and so on. Since the routes have been stored this will save us from recalculating the routes again.

Failure of vice cluster head. On the instance of vice cluster head failure, we go back to the network setup phase, recluster the network, and determine new routes. The fault tolerance algorithm has been given in Algorithm 2.

25.6 Simulation Results and Discussion

25.6.1 Experimental Setup

The proposed protocol was simulated using MATLAB R2015a on an Intel i5 machine with 2.40 GHz and 16 GB RAM running on Ubuntu 15.10. We deployed

Algorithm 1 Proposed routing algorithm

```
Input: $\forall CH_k,\ Energy_k,\ Distance\ to\ Sink_k$
Output: $All\ paths,\ from\ Source\ CH_i\ to\ Sink$
procedure CH–ROUTESELECTION
    $Node_i$ receives $REQ_i$ packet from it $NeighborNode(i)$
    if $Node_i! = sink$ then
        if $Energy(i) < E_thresh$ then
            $REQ_i \leftarrow REQ_i + (Id_i, Energy_i, Dist_i)$
            Forward $REQ_i$ to $NeighborNodes(i)$
        else
            $Node_i$ does not broadcast
        end if
    else if $Node_i$ is == Sink then
        Start $timer_j$
        while $timer_i < Time_{thresh}$ do
            if $REQ == REQ_i$ then
                $REQSet(i) \leftarrow REQSet(i) \cup REQ_i$
            end if
        end while
        for each Request $REQ_i \in REQSet(j)$ do
            $Cost(j,i) \leftarrow 0.3\ \times Dist(Source, Sink) + 0.7\ \times Energy(Source, Sink)$
        end for
        for each row $Cost(j,:)$ do
            Sort $Cost(j,:)$
            Set Priority in Descending Order in $Cost(j,:)$
        end for
        for each $REQ_i$ in $REQSet(j)$ do
            $REP_i \leftarrow REP_i + (Id_i, Path_i, Priority_i)$
            Forward $REP_i$ to $Source_i$
        end for
        Node $N_i$ creates routing table using REP messages.
    end if
end procedure

procedure CH–ROUTING
    Use the Path with $Priority = 1$
end procedure
```

400 sensor nodes in a square area of size 300×300 square meters. The topology of the simulated network is illustrated in Fig. 25.2. The sensor nodes were considered to have a starting energy of 2 J. When the energy level of the node reached 0 J the node was considered dead. We use Weibull reliability function [10] to model the faults in the cluster heads in our network which is given by

$$R(t) = e^{-\left(\frac{t-\gamma}{\eta}\right)^{\beta}} \tag{25.5}$$

where γ is the location parameter, η is the scale parameter, and β is the shape parameter. We set the values of $\gamma = 0$, $\beta = 3$, and $\eta = 3000$. If β is greater than 1, then the rate of failure increases with time else if β is less than 1, then the rate of failure decreases with time. Moreover, $\beta = 0$ the rate of failure is constant.

Algorithm 2 Fault tolerance algorithm

```
procedure CH–FAULTTOLERANCE
    for $i \in$ Priority do
        if Path with $Priority = i$ fails due to CH failure then
            if Vice CH not used then
                Awake Vice CH
                Replace CH with Vice CH
                Update routing Tables
            end if
        else
            Use path with $Priority = i + 1$
        end if
    end for
end procedure
```

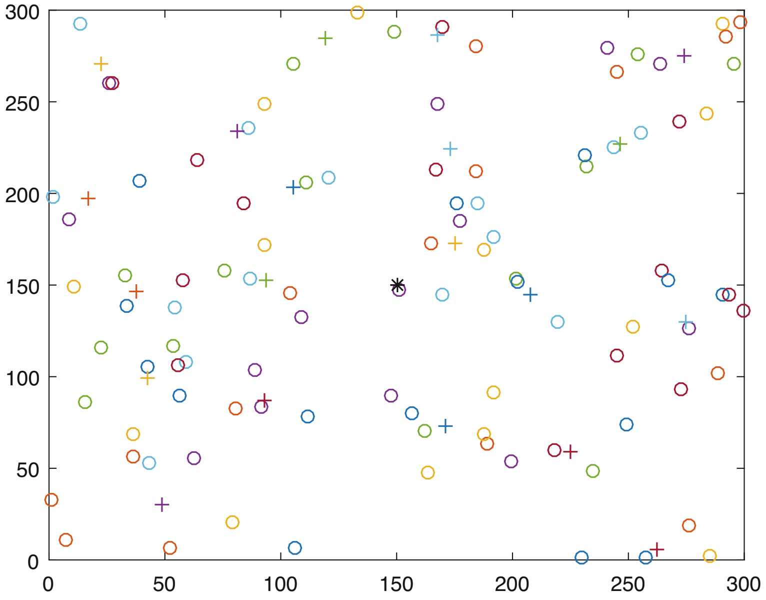


Fig. 25.2 The simulated network topology

β is chosen as per the analysis provided in [31] where it is established that the failure of cluster heads can be represented using a Weibull distribution with $\beta = 3$. Furthermore, [12] uses $\beta = 3$ for gateway faults in WSN. The simulation parameters used in the simulation are shown in Table 25.5. The parameters used are similar to [25].

The proposed algorithm is compared with DFTR [12] and LEACH [14] in terms of residual energy, number of packets received at sink, number of dead cluster heads, and network lifetime. We discuss the results of the experiments in the following sections.

Table 25.5 The simulation parameters

Simulation parameters	
Network size	400
Number of clusters	300×300
Initial sensor node energy	2.0 J
E_{elec}	50 nJ/bit
E_F	5 nJ/bit
Communication range	100 m
ε_{FS}	10 pJ/bit/m^2
ε_{MP}	0.0013 pJ/bit/m^4
d_0	88 m
Packet size	4000 bits
Message size	200 bits
E_{thresh}	20%

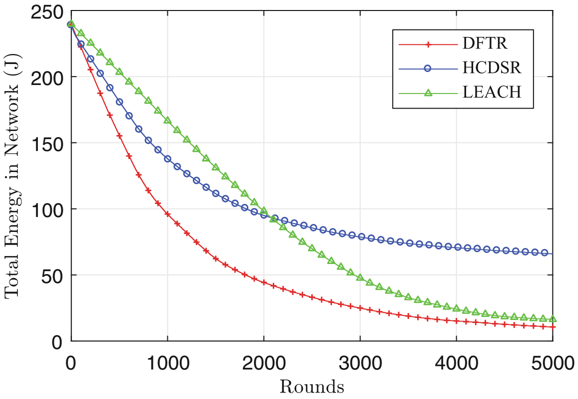


Fig. 25.3 The number of alive nodes per round in the simulated network among DFTR, LEACH, and HCDSR

25.6.2 Analysis of HCDSR

A wireless sensor network consisting of 120 nodes was simulated and clustered initially into 20 clusters. These 120 nodes were deployed in a sensing field of size 300 × 300. The sink was placed at the center of the sensing field at the coordinates (150,150). The simulated network is depicted in Fig. 25.2. The total number of alive nodes is compared in Fig. 25.3. We can see that the total alive nodes after 5000 rounds for the proposed technique are more than the LEACH and DFTR. Nodes are considered dead when their energy reaches 0 J or due to the simulation of faults following the Weibull distribution. We can observe in our proposed technique that initially there is a decrease in alive nodes that stabilize after a certain amount of rounds.

The stability of alive nodes is due to the energy threshold that was applied which resulted in many cluster heads with lower energy not to participate in the clustering, whereas in LEACH protocol, we can observe that the rate of dead nodes increases after a certain number of rounds. The DFTR protocol that does not provide an energy threshold has the least amount of total energy in the network. This is because once the cluster head dies in DFTR technique and nodes of clusters become orphan they have to send it to a longer distance. Since DFTR uses special fixed gateways, a failure in the cluster head means another normal node doesn't take its place as a replacement, unlike the proposed technique. Figure 25.4 shows the total energy in the network per round for each of the technique. It is also similar to the previous graph where the total energy at the end of 5000 rounds is highest in the proposed technique. Hence, we can clearly say that the proposed technique increases the overall lifetime of the network as compared to LEACH and DFTR.

The total number of packets that have been transmitted to the sink for 5000 rounds has been compared in Fig. 25.5. We can see that the proposed technique transmits the maximum amount of packets. This is due to the presence of vice cluster heads which replace the failed cluster heads unlike LEACH protocol or DFTR protocol. A higher number of packets transmitted to the base station indicates the longer life of gateways. We can see that the proposed algorithm performs better than both DFTR and LEACH.

Fig. 25.4 Comparison of total energy in the simulated network per round among DFTR, LEACH, and HCDSR

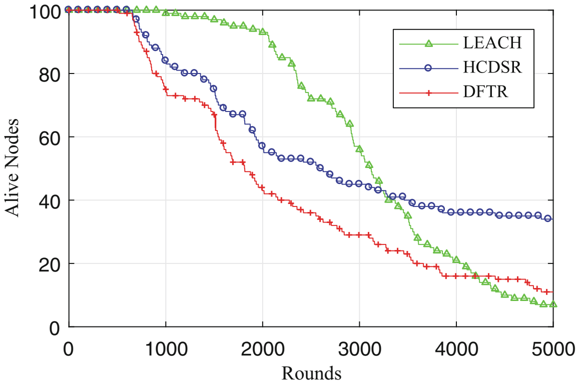
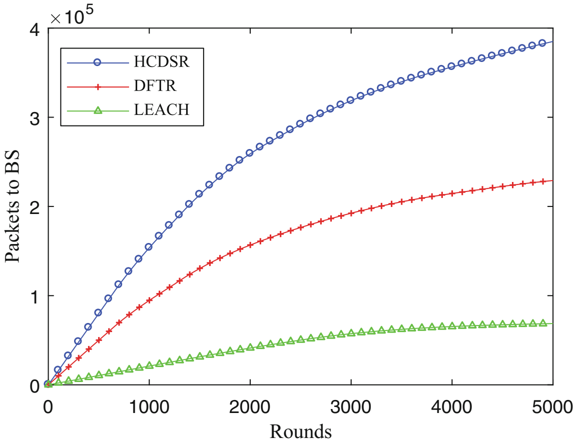


Fig. 25.5 The total number of data packets received at sink per round among DFTR, LEACH, and HCDSR



m.thaha.h@ieee.org

25.7 Conclusions

In this article, we have proposed a reliable and resilient routing technique for wireless sensor networks that forms the atomic component of IoT for smart city applications. We have proposed a taxonomy for fault tolerant techniques in WSN. Furthermore, we proposed a new fault tolerant routing algorithm for hierarchical WSN networks based on modified DSR (dynamic source routing) and vice cluster heads. Multiple routes are identified and these routes are prioritized on the basis of residual energy in the path and the distance of the source from the sink. In addition, the proposed technique uses vice cluster heads to tolerate faults during routing. We have shown through simulation that the proposed technique is better than LEACH and DFTR in terms of total energy in the network, the total number of packets transmitted to the sink, and the number of alive nodes. Our future work will be based on the mobility of the sensor nodes and the interaction with other IoT components.

Acknowledgements The authors acknowledge with thanks the technical and financial support from the Deanship of Scientific Research (DSR) at the King Abdulaziz University (KAU), Jeddah, Saudi Arabia, under the grant number G-651-611-38. The work carried out in this paper is supported by the High Performance Computing Center at the King Abdulaziz University, Jeddah.

References

1. Abbasi, A.A., Younis, M.: A survey on clustering algorithms for wireless sensor networks. Comput. Commun. **30**(14–15), 2826–2841 (2007)
2. Ahmed, R.E.: A fault-tolerant, energy-efficient routing protocol for wireless sensor networks. In: 2015 International Conference on Information and Communication Technology Research (ICTRC), pp. 175–178 (2015)
3. Akkaya, K., Younis, M.: A survey on routing protocols for wireless sensor networks. Ad Hoc Netw. **3**(3), 325–349 (2005)
4. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. IEEE Commun. Mag. **40**(8), 102–114 (2002)
5. Alam, F., Mehmood, R., Katib, I., Albeshri, A.: Analysis of eight data mining algorithms for smarter internet of things (IOT). Procedia Comput. Sci. **98**(Suppl. C), 437–442 (2016). http://www.sciencedirect.com/science/article/pii/S187705091632213X
6. Alam, F., Mehmood, R., Katib, I., Albogami, N.N., Albeshri, A.: Data fusion and IoT for smart ubiquitous environments: a survey. IEEE Access **5**, 9533–9554 (2017)
7. AlTurki, R., Mehmood, R.: Multimedia ad hoc networks: performance analysis. In: 2008 Second UKSIM European Symposium on Computer Modeling and Simulation, pp. 561–566 (2008)
8. Alturki, R., Mehmood, R.: Cross-layer multimedia QoS provisioning over Ad Hoc networks. In: Using Cross-Layer Techniques for Communication Systems, pp. 460–499. IGI Global, Hershey (2012). http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-4666-0960-0.ch019
9. Anastasi, G., Conti, M., Di Francesco, M., Passarella, A.: Energy conservation in wireless sensor networks: a survey. Ad Hoc Netw. **7**(3), 537–568 (2009)
10. Antle, C.E., Bain, L.J.: Weibull distribution. In: Encyclopedia of Statistical Sciences, vol. 12, pp. 7629–7634. Wiley, Hoboken (2004)

11. Arfat, Y., Aqib, M., Mehmood, R., Albeshri, A., Katib, I., Albogami, N., Alzahrani, A.: Enabling smarter societies through mobile big data fogs and clouds. Procedia Comput. Sci. **109**, 1128–1133 (2017). http://www.sciencedirect.com/science/article/pii/S1877050917311213
12. Azharuddin, M., Jana, P.K.: A distributed algorithm for energy efficient and fault tolerant routing in wireless sensor networks. Wirel. Netw. **21**(1), 251–267 (2015)
13. Azharuddin, M., Jana, P.K.: A PSO Based Fault Tolerant Routing Algorithm for Wireless Sensor Networks, pp. 329–336. Springer, New Delhi (2015)
14. Balakrishnan, H., Heinzelman, W.R., Chandrakasan, A.: Energy-efficient communication protocol for wireless microsensor networks. In: 2014 47th Hawaii International Conference on System Sciences, vol. 08, 8020 (2000)
15. Bogliolo, A., Lattanzi, E., Acquaviva, A.: Energetic sustainability of environmentally powered wireless sensor networks. In: Proceedings of the 3rd ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor and Ubiquitous Networks, PE-WASUN '06, pp. 149–152. ACM, New York (2006)
16. Bottero, M., Chiara, B.D., Deflorio, F.: Wireless sensor networks for traffic monitoring in a logistic centre. Transp. Res. C Emerg. Technol. **26**, 99–124 (2013)
17. Boucetta, C., Idoudi, H., Saidane, L.A.: Adaptive scheduling with fault tolerance for wireless sensor networks. In: Vehicular Technology Conference (VTC Spring), 2015 IEEE 81st, pp. 1–5. IEEE, Piscataway (2015)
18. Calero, C., Caro, A., Piattini, M.: An applicable data quality model for web portal data consumers. World Wide Web **11**(4), 465–484 (2008)
19. Casey, K., Lim, A., Dozier, G.: A sensor network architecture for tsunami detection and response. Int. J. Distrib. Sens. Netw. **4**(1), 27–42 (2008)
20. Cetinkaya, O., Akan, O.B.: Use of wireless sensor networks in smart homes. In: Emerging Communication Technologies Based on Wireless Sensor Networks: Current Research and Future Applications, pp. 233–258 (2016)
21. Gelenbe, E., Ngai, E.: Adaptive random re-routing for differentiated QOS in sensor networks. Comput. J. **53**(7), 1052–1061 (2010)
22. Gillies, D., Thornley, D., Bisdikian, C.: Probabilistic approaches to estimating the quality of information in military sensor networks. Comput. J. **53**(5), 493–502 (2010)
23. Gupta, S.K., Kuila, P., Jana, P.K.: E3bft: energy efficient and energy balanced fault tolerance clustering in wireless sensor networks. In: 2014 International Conference on Contemporary Computing and Informatics (IC3I), pp. 714–719 (2014)
24. Hamdan, D., Aktouf, O.E.K., Parissis, I., El Hassan, B., Hijazi, A.: Integrated fault tolerance framework for wireless sensor networks. In: 2012 19th International Conference on Telecommunications (ICT), pp. 1–6. IEEE, Piscataway (2012)
25. Heinzelman, W.B., Chandrakasan, A.P., Balakrishnan, H.: An application-specific protocol architecture for wireless microsensor networks. IEEE Trans. Wirel. Commun. **1**(4), 660–670 (2002)
26. Hezaveh, M., Shirmohammdi, Z., Rohbani, N., Miremadi, S.G.: A fault-tolerant and energy-aware mechanism for cluster-based routing algorithm of WSNs. In: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 659–664 (2015)
27. Khan, S.A., Bölöni, L., Turgut, D.: Bridge protection algorithms a technique for fault-tolerance in sensor networks. Ad Hoc Networks **24**, 186–199 (2015)
28. Kimençe, Ş., Bekmezci, İ.: Weighted relay node placement for wireless sensor network connectivity. Wirel. Netw **20**(4), 553–562 (2014)
29. Kuila, P., Jana, P.K.: Improved load balanced clustering algorithm for wireless sensor networks. In: Thilagam, P.S., Pais, A.R., Chandrasekaran, K., Balakrishnan, N. (eds.) Proceedings of the 2011 International Conference on Advanced Computing, Networking and Security, ADCONS'11, pp. 399–404. Springer, Berlin (2012)
30. Kuila, P., Jana, P.K.: Approximation schemes for load balanced clustering in wireless sensor networks. J. Supercomput. **68**(1), 87–105 (2014)

31. Lee, J.J., Krishnamachari, B., Kuo, C.C.J.: Aging analysis in large-scale wireless sensor networks. Ad Hoc Netw. **6**(7), 1117–1133 (2008)
32. Li, Y., Xiao, G., Singh, G., Gupta, R.: Algorithms for finding best locations of cluster heads for minimizing energy consumption in wireless sensor networks. Wirel. Netw. **19**(7), 1755–1768 (2013)
33. Mehmood, R., Alturki, R.: A scalable multimedia QoS architecture for ad hoc networks. Multimed. Tools Appl. **54**(3), 551–568 (2011). https://doi.org/10.1007/s11042-010-0569-0
34. Mehmood, R., Alturki, R.: Video QoS analysis over Wi-Fi networks. In: Advanced Video Communications over Wireless Networks, pp. 439–480. CRC Press, Boca Raton (2013)
35. Mehmood, R., Nekovee, M.: Vehicular ad hoc and grid networks: discussion, design and evaluation. In: Proceedings of the 14th World Congress on Intelligent Transport Systems (ITS), Beijing (2007)
36. Mehmood, R., Alturki, R., Faisal, M.: A scalable provisioning and routing scheme for multimedia QoS over ad hoc networks. In: Mauthe, A., Zeadally, S., Cerqueira, E., Curado, M. (eds.) Future Multimedia Networking, pp. 131–142. Springer, Berlin (2009)
37. Mehmood, R., Alam, F., Albogami, N.N., Katib, I., Albeshri, A., Altowaijri, S.M.: Utilearn: A personalised ubiquitous teaching and learning system for smart societies. IEEE Access **5**, 2615–2635 (2017)
38. Morello, R., Mukhopadhyay, S.C., Liu, Z., Slomovitz, D., Samantaray, S.R.: Advances on sensing technologies for smart cities and power grids: a review. IEEE Sens. J. PP(99), 1–1 (2017)
39. Muhammed, T., Shaikh, R.A.: An analysis of fault detection strategies in wireless sensor networks. J. Netw. Comput. Appl. **78**(Suppl. C), 267–287 (2017). http://www.sciencedirect.com/science/article/pii/S1084804516302545
40. Muhammed, T., Mehmood, R., Albeshri, A.: Enabling reliable and resilient IOT based smart city applications. In: Mehmood, R., Bhaduri, B., Katib, I., Chlamtac, I. (eds.) Smart Societies, Infrastructure, Technologies and Applications, pp. 169–184. Springer, Cham (2018)
41. Nations, U.: Global Urban Observatory (GUO) UN-Habitat, https://unhabitat.org/urban-knowledge/guo/
42. Nikam, S.S., Mane, P.B.: Swarm Intelligent WSN for Smart City, pp. 691–700. Springer, Singapore (2017)
43. Pantazis, N., Nikolidakis, S.A., Vergados, D.D.: Energy-efficient routing protocols in wireless sensor networks: a survey. IEEE Commun. Surv. Tutorials **15**(2), 551–591 (2013)
44. Ramanathan, N., Kohler, E., Girod, L., Estrin, D.: Sympathy: a debugging system for sensor networks [wireless networks]. In: Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks, pp. 554–555. IEEE Computer Society, Washington, DC (2004)
45. Ringwald, M., Römer, K., Vitaletti, A.: Snif: sensor network inspection framework. Tech. Rep. 535, Department of Computer Science, ETH Zurich, Zurich (2006)
46. Shnayder, V., Hempstead, M., Chen, B.R., Allen, G.W., Welsh, M.: Simulating the power consumption of large-scale sensor network applications. In: Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems, SenSys ’04, pp. 188–200. ACM, New York (2004)
47. Wang, R., Zhang, L., Sun, R., Gong, J., Cui, L.: Easitia: a pervasive traffic information acquisition system based on wireless sensor networks. IEEE Trans. Intell. Transp. Syst. **12**(2), 615–621 (2011)
48. Xu, J., Liu, W., Lang, F., Zhang, Y., Wang, C.: Distance measurement model based on RSSI in WSN. Wirel. Sens. Netw. **02**(08), 6 (2010)
49. Zhao, Y., Wu, J., Li, F., Lu, S.: On maximizing the lifetime of wireless sensor networks using virtual backbone scheduling. IEEE Trans. Parallel Distrib. Syst. **23**(8), 1528–1535 (2012)