

Research Article

SALMA: An Efficient State-Based Hybrid Routing Protocol for Mobile Nodes in Wireless Sensor Networks

Muhammad Muneer Umar,¹ Nabil Alrajeh,² and Amjad Mehmood¹

¹*Institute of Information Technology, Kohat University of Science & Technology, Kohat 26000, Pakistan*

²*Biomedical Technology Department, College of Applied Medical Sciences, King Saud University, Riyadh 11451, Saudi Arabia*

Correspondence should be addressed to Nabil Alrajeh; nabil@ksu.edu.sa

Received 30 November 2015; Revised 18 January 2016; Accepted 24 January 2016

Academic Editor: Luis Javier Garcia Villalba

Copyright © 2016 Muhammad Muneer Umar et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the rapid development in mobile Wireless Sensor Networks (WSNs), it has become very essential to focus on the efficiency in performance of small sensing nodes operating in WSNs. While designing a routing protocol for mobile sensor nodes, the quality parameters like end-to-end delays and routing overhead are always considered. Moreover, the nodes in wirelessly connected mobile networks consume considerable power on routing more than other functionalities. Any modification in a standard routing protocol can also affect routing overhead, end-to-end delays, and energy consumption of nodes. In this paper a new hybrid routing protocol, named as State-Aware Link Maintenance Approach (SALMA), is introduced which is based on Dynamic Source Routing (DSR) and Optimized Link State Routing (OLSR) protocols. The work also focuses on the activeness of nodes in the network operations and defines three states of nodes, that is, white, gray, and black. The work concludes that the proposed protocol gives improvements in some quality of service metrics like lower delay than DSR, lower routing overhead than OLSR, and lesser energy consumption by the network nodes.

1. Introduction

Wireless Sensor Networks (WSNs) are being deployed for various purposes in a wide range of critical applications such as environmental monitoring, health-care, and military applications. Usually nodes in such environments are densely installed to receive important sensed data for precise monitoring. Similarly, the response time is very critical performance metric in such intrusion detection application. The sensed data must be delivered in no time, in case any instruction is detected in the environment. It is always desired to have a mechanism for keeping nodes alive for a longer period to be able for transmitting their sensed data [1]. Beside the energy factor, sensor nodes are usually installed in harsh environments, which results in nodes' damage and failure very commonly. Therefore implemented routing protocols must be able to prolong the life of a network along with handling the dynamic topological changes.

In most of the wireless networks with point-to-point links, some procedures are essential to be defined for routing the packets from source to target nodes. The routing comprises two main protocols, the route discovery and route maintenance, beside associated costs. It is always assumed that all the nodes must collaborate with others [2]. Unlike other systems with dedicated routers, in mobile WSNs all the nodes have scarce computational resources and, more significantly, they operate on very limited battery power. Developing a routing choice in a dynamically changing network always requires route maintenance procedure [3].

Routing protocols are very critical for the basic operations of mobile nodes in MANETs and WSNs. Development of an efficient routing protocol for such networks is significantly a challenging assignment. In such networks nodes collaborate and forward data packets to one another through different links [4]. Rapid topological changes due to continuous mobility of nodes cause frequent routes and data packets

loss. The protocols in such networks must allow the nodes to configure themselves automatically [5]. The performance of limited powered nodes and shared wireless medium with variable bandwidth for communication can be effected by the routing control overhead on the nodes [6].

Proactive protocols [7] keep consistent and up-to-date fresh links to each other nodes. Any update in topological structure is propagated all over the network in order to maintain a probable unfailling route information in a network. Such protocols periodically update their routing tables causing immediate selection of a forward node to minimize initial delays in the data transfer. However the excessive generation of control traffic incurs communication overhead causing extra consumption of scarce resources of nodes. Particularly in an environment of highly mobile nodes proactive algorithms can be very costly at the peak.

Reactive protocols [7] are on-demand routing protocols in the sense that routes are found and established only when a source node initiates to send data packets to a destination node. When a node, the source, or initiator needs a route to the destination or target node, it initiates a route discovery protocol by flooding a Route Request (RREQ) packet throughout the network. In this process the intermediate nodes also read the route information from this packet. The destination sends the source node a Route Reply (RREP) packet to establish the route. A maintenance procedure is also used to keep the latest route information. Comparing with the proactive routing protocols nodes use very less bandwidth for maintenance of routing tables in these protocols by avoiding excessive timely generation of routing information updates. The drawback of such protocols is that the route discovery latency critically increases which leads to delay in transferring the data packets or starting communication between nodes.

Hybrid routing protocols [8] is an effort to combine the best characteristics of both proactive and reactive routing protocols. These protocols divide network into different segments known as zones [9] considering some criteria. Proactive algorithms are operated inside the zone while reactive routing protocols are used for getting route information from other zones. Another form of hybrid protocols is hierarchical protocols [10] which operate both proactive and reactive routing protocols on different hierarchic levels of a network in which a node resides. At the top of hierarchy the proactive routing is established and then the route discovery in lower levels is performed reactively. The choice for reactive or proactive routing protocol requires suitable identification for individual levels. It has a flaw that usually response to traffic request relies on meshing parameters in most of the cases.

State-Aware Link Maintenance Approach (SALMA) combines both the reactive and proactive routing protocols to reduce the overhead on most of the nodes and increases network performance by reducing the load of network discovery flooding on active nodes. This protocol divides the network nodes into three categories: (a) aware and active nodes known as *black node*, (b) aware but not performing data transfer except data forwarding which are called *gray nodes*, and (c) *white nodes* which are idle and do not keep any routing information. SALMA uses a data structure *Keep*

Awake Buffer, explained in the next sections, to determine the type of these nodes. The approach operates reactively in the beginning of any setup. Once a node starts its operations the route is maintained proactively to reduce the flooding of control packets for route discovery.

Moreover SALMA utilizes the statistics of usage of a node. If a node is continuously nonactive, it is kept idle and can be revoked by either forwarding packet or its own functionality. An active node periodically detects its neighbors to update the route information. Like Optimized Link State Routing (OLSR) [11] links and neighbors are sensed by generating a HELLO message periodically. The idle nodes do not respond to the HELLO messages in order to reduce the consumption of their limited resources. This protocol is highly influenced by Dynamic Source Routing (DSR) [12] protocol for route discovery procedure and somehow in route maintenance. Route is maintained primarily proactively by adopting some rudiments of OLSR protocol.

The work comprises two majors: the study of developed routing protocols and the proposed approach for routing in mobile WSNs. This work is structured as follows: we elaborate the related routing protocols in Section 2 and then in Section 3 explain the SALMA protocol. Lastly in Section 4 the simulation results are discussed and Section 5 includes the conclusion with future work.

2. Related Work

When sensor nodes in a WSN are not mobile then it is better to have proactive routing protocols like OLSR and Destination-Sequenced Distance-Vector (DSDV). Because reactive routing protocols, like DSR and Ad Hoc On-Demand Distance Vector (AODV), consume significantly higher amount of energy in discovering and setup of routes. Moreover, reactive protocols sometime in case of a large network may produce considerable initial delays while initiating route discoveries. In case of mobile nodes, the route control messages in proactive routing algorithms may not be effective and are challenging due to continuous topological changes [13].

In proactive protocols the network status is periodically updated which results in low latency and requires extra bandwidth while reactive routing protocols operate in bursty style, which may cause congestion in the network during higher traffic. During inactivity period these protocols save energy and bandwidth [14]. DSR protocol is very efficient for use in mobile nodes of multihop Wireless Sensor Networks. This protocol, as compared to AODV, has significant larger amount of routing data as it allows more paths from source to a destination. DSR gives better performance by having access to multiple alternate routes which reduces the load of flooding route discoveries [15]. However the reactive nature of DSR may affect the performance of entire network in various situations.

There have been some famous hybrid protocols, such as Zone Routing Protocol (ZRP) [9] and Sharp Hybrid Adaptive Routing Protocol (SHARP) [10]. ZRP uses an already stored routing table for communication with a node inside the same

zone of the source to deliver the packets without any delay. To check the availability of target node in another zone, the source node initiates a route discovery reactively to reduce the processing overhead. Once zone is confirmed for having the target node, the source node uses routing table for data packet transfer proactively [9]. SHARP automatically finds the optimal point of application of both the reactive and proactive routing protocols. It adjusts the level of routing information to be transferred proactively and the level of reactively discovery of the routes. This protocol is application specific for minimizing packet overhead, controlling jitter and bounding loss rate [10].

Another famous hybrid routing protocol based on Ant Colony Optimization (ACO) and ZRP for MANETs is known as HOPNET. This algorithm solves routing problems with Swarm Intelligence, a subarea of artificial intelligence [16]. The protocol adopts the same mechanism of zones like ZRP. The intrazone communication is made proactively, while for interzone communication the protocol uses reactive route discovery.

In mobile WSNs there are always various restrictions like energy consumption and processing power, and so forth. Nodes in WSNs usually operate on nonrechargeable batteries; therefore routing algorithms for these networks must focus on the energy efficiency. In this context the pure hybrid routing protocols like HOPNET and ZRP are not appropriate for WSNs. The primary challenges of routing in WSNs are to prolong the lifetime of the network and improve energy efficiency in all the nodes along with support for data communication and control [17]. The mobility of nodes usually effect the topological structure of zones in the network. Therefore updating information regarding these dynamic zones itself is a challenge for implementing typical zone-based hybrid routing protocol in mobile WSNs.

A remarkable work has been done for making efficient hybrid routing protocols for networks with sensing and mobile nodes. A hybrid routing protocol inspired from ZRP with dynamic zones is introduced for WSNs in [18]. It is designed to handle the overhead of dynamic topologies. In this work the authors combine ZRP and ACO to improve the quality of a network with dynamic sized zones. The proposed work minimizes the latency while reducing the network overhead. To reduce complexity the protocol uses only one routing table for both inter- and intrazone communication.

Some authors proposed the adaptation of HOPNET protocol in pure WSNs considering the critical issues of limited energy, processing power, and memory of nodes [19]. They presented some solutions to update the HOPNET protocol to increase its efficiency and adaptability for decreasing the routing complications in WSNs. The suggested modifications include the elimination of interzones routing table, simplified ants, and changed route discovery mechanism.

Another protocol named as Simple Hierarchical Routing Protocol (SHRP) discussed in [20, 21] uses different metrics to maintain the reliable data delivery. According to authors the transmission of data consumes more energy; therefore it is better to save energy by not sending all the periodic data to the sink. The coordinator nodes save the energy by combining various data messages into one for further sending it to the

destination. The protocol also delivers the load balancing facility. This protocol periodically updates the routing tables by checking the nodes contributions along with the battery lifetime of nodes.

In Distributed Spinning Trees (DST) [22] network nodes are divided into three states, that is, router, merge, and configure. The network is structured into various trees. Each tree keeps internal and router nodes. The nodes distribution is made for control purpose only and does not address energy efficiency in ad hoc networks.

Al-Rokabi and Politis [23] propose a hybrid routing protocol for MANETS. They named it as "Hybrid Hop count based Multiple Quality of Service (QoS) constraints Routing Protocol with Mobility Prediction (HMQRMP)". In this work a node can behave either reactively or proactively depending on its energy level. Moreover the paper also focuses on different QoS parameters like bandwidth, delay-jitter, and hop count. It deals with QoS parameters of wireless links, namely, delay, delay-jitter, bandwidth, cost, link expiry time, and residual battery power of mobile nodes. HMQRMP selects the best routing path with minimal hop count, maximal link expiry time, and high energy level among multiple paths between a source and a destination as to increase packet delivery ratio and reduce control overhead in MANET. It predicts the stability of link expiry time of each link for every path. HMQRMP changes the behavior of a mobile node from proactive to reactive or vice versa by comparing the nodes energy level against the power thresholds so as to be adoptable to topological changes which increases the lifetime of mobile nodes in MANET.

Simplified OLSR and AODV Protocol (SOAP) presented in [24] is a hybrid protocol having both the proactive and reactive features. The protocol uses a modified version of OLSR routing protocol with AODV routing protocol. A logical layer is defined which chooses OLSR for data transmission, if the source nodes has route information else it uses AODV for the route discovery. The proposed work claims to produce improved performance in terms of lower overhead and reduced delays, but it does not reflect the energy efficiency in the whole network.

In [25], an energy-aware routing protocols based on the combination of planar routing and hierarchical routing is proposed. The nodes' state and the route selection is determined according to the residual energy of nodes. The work uses cluster based WSN with cluster heads responsible for data communication through them. The work emphasizes on the delivery rate, the energy consumption, and the efficiency in the network. However, it does not address the participation and activeness of individual nodes in the whole functionality of the network.

In [26] authors propose a scheme by using subideal routes infrequently to provide considerable gains. Multiple routes are searched from source and destination nodes, and each route is allocated a probability of being selected, based on the energy metric. Routes are chosen randomly all the time due to their random probability dependent selection which ultimately prevents energy reduction.

The authors in [27] propose a modified version of OLSR protocol. They present a different mechanism for selection

of the Multipoint Relays (MPRs) based on the concept of energy-aware willingness. Each node presents its default willingness to become an MPR. Devi [28] integrates the AODV protocol with threshold-sensitive energy efficient sensor network (TEEN) protocol and formed a new hybrid AODV routing protocol. The main focus in this work is to enhance energy of the network by determining the active nodes. Moreover, the authors also address the efficiency in packets transmission, throughput, and reduced latency.

Many authors proposed schemes for load-balancing and reduced burden on some targeted nodes. In [29], the authors propose a QoS-aware multipath hybrid routing protocol. The routes are discovered reactively while the topology discovery is made in proactive way. Topology Information Table (TIT) is used on each node to store the topological information like battery power, residual bandwidth, and so forth. Multipath Dijkstra algorithm is used to establish multiple paths with the help of reactive route discovery method. The work puts an extra burden of topology discovery and increases the energy consumption of all the nodes in the network.

Another research proposed in [17] uses load-balancing with adaptive threshold routing algorithm for ad hoc networks. A reactive method is used for dividing the traffic load consistently among various nodes of the networks. During route discovery, if a node is overloaded then it is allowed to drop the packets. The work reduces the system performance in many aspects.

Numerous researches proposed efficient routing protocols by introducing protocols for cluster based WSNs. LEACH [30], a cluster based routing protocol, uses cluster heads to distribute the power usage burden among the sensor nodes in the WSN. The cluster heads are randomly rotated to balance the entire load just like other energy-aware protocols for routing in WSNs. The work relies on cluster based network and does not address the state of individual nodes. Moreover, cluster heads are directly connected by each node to transmit data. Therefore, it is not suitable for mobile nodes deployed in a large range.

Another hybrid routing protocol, proposed in [31], establishes routes proactively at earlier stages. Then the routes are managed reactively to reduce the periodic data gathering. The protocol is cluster based and highly inspired from LEACH. The network is divided into two hierarchical levels to reduce the number of intermediate hops towards the sink. For energy efficiency the protocol uses procedures performed by the cluster heads. However it does not utilize the hybrid nature for energy efficiency in the network.

For efficient consumption of nodes' energy, a method of creating sensor nodes into a chain is defined in PEGASIS [32]. For multihop data transmission, sensor nodes select nearer nodes in the cluster rather than CH. Some authors introduced a Hierarchical Routing Protocol (HRP) with a purpose to extend the network's life [33]. A multihop communication protocol, proposed in [34], introduces a coordinator node in the cluster. An extension of LEACH, pLEACH protocol [35], defines the procedure of CHs selection according to their highest remaining residual energy.

3. State-Aware Link Maintenance Approach (SALMA)

The main theme of this work is to categorize the mobile nodes in a WSN into three states. These states of nodes function differently for the sake of minimizing the activity load on nodes. In the first phase, SALMA protocol develops the routing tables stored on nodes reactively. The nodes keep their activity status and can change their categories reflecting their behavior in the network. Once an active node updates its routing table, it maintains its routes proactively to lessen the delays in transmission of data packets. The nodes not involved in data transmission are kept somehow nonactive to reduce their resource consumption. The work is divided into various subsections elaborated below.

3.1. Data Structures. Various types of data structures are used in SALMA. These data structures are essential for implementation of functionality of SALMA protocol. The data structures are explained as follows.

- (a) *Routing Table.* Each nonwhite node maintains its routing table to know different paths for data transfer to destination nodes. The routing tables are updated if a node finds a new route by reading a passing packet or detects a link break. Entries in the routing table are added or modified by the intermediate nodes if they discover new or short routes by reading the data and control packets. If a node finds a route broken due to nonreply of HELLO or on receiving route error, then it deletes the route information from the table.
- (b) *Keep Awake Buffer (KAB).* This buffer keeps a numerical value to identify the state and operation frequency of a node. The value of KAB is triggered by a timer known as Keep Awake Counter (KAC). A white node keeps 0 in its KAB. However gray and black nodes are identified by odd and even numbers in their KABs, respectively. The higher the number stored in KAB is, the higher the frequency of node's activeness is presented.
- (c) *Keep Awake Counter (KAC).* It is a simple integer type data structure used to decrement the value of KAB after a specific period of time. The KAC operates on system's clock and decreases its value to 0. Upon reaching to zero the KAC decrements the value of KAB and resets its own value to the maximum. Its maximum value can be set on the basis of the environment in which this protocol is used for any mobile WSN. The higher value will cause less transformation of nodes' states.
- (d) *Route Request Queue.* Each RREQ is stored in a queue before generation. The RREQ is automatically deleted from the queue once a route is discovered. In case of nonavailability of route, a node can keep this queue for future attempts on route discovery.
- (e) *Send Queue.* On each new entry in the routing table nodes scan the entries in their send queue to check the availability of queued data packets to be transferred.

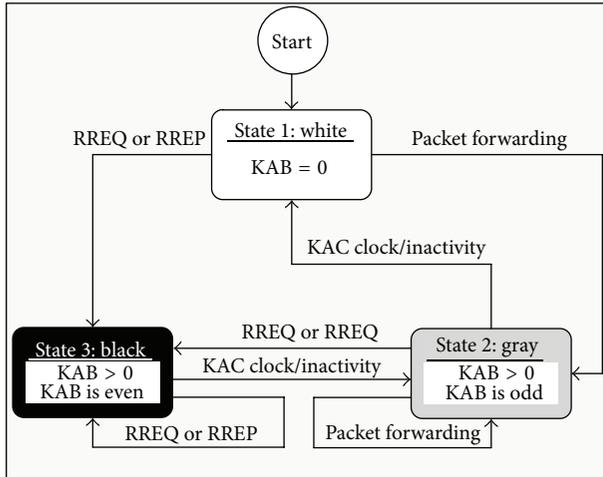


FIGURE 1: Nodes' state transition diagram.

Once the route is established the entries from this queue are deleted after successful transmission. A white node does not keep information in its send queue.

- (f) *Neighbors' States Array (NSA)*. With reception of each reply to HELLO messages this queue is altered. The nodes reply with HELLO messages with their KAB to the senders. On receiving a reply of HELLO message, nodes save the acknowledged KAB of the sender. The entries in this array additionally keep the storage time field. KAC also influence the value of this storage by deleting the older entries.

3.2. Classification of Nodes. In this approach nodes in the whole network are categorized into three states: (a) white nodes, (b) gray nodes, and (c) black nodes. The motivation for this labeling is to reduce the route maintenance load on most of the sleeping or idle nodes.

All these nodes continuously convert their states according to their activities. White and gray nodes can vary to any type except the black nodes which can be changed to only gray nodes. The state diagram, in Figure 1, shows the switching of nodes from one state to another. A KAB is used to determine the state of a node. The three categories are explained as follows:

- (a) *White Nodes*. These nodes do not keep the latest route information in their routing tables. In most of the cases a node can become white due to its idleness and due to its nonfunctionality or nonavailability of route, for a specified period of time. These nodes are revoked by their own packet transmission initialization, a RREQ from the source for data transfer and their neighbors' packet forwarding request. If a white node intends to transmit data or receives a RREQ, then it develops to a black node. Once a node becomes black it must update the routing table with the new route information, increment the value of KAB by two,

and activate the KAC. In further case of forwarding packets, it is converted into a gray node. By receiving a packet for forwarding the node reads the route information from the packet header and forwards it to the next hop. Whenever a white node converts to a gray, it must increment the value of KAB by one, update its routing table with new entries and activate the KAC before conversion. By receiving a HELLO message from a neighbor these nodes, due to long idleness and having unusable routing table, do not reply for the purpose of saving their resources.

- (b) *Gray Nodes*. These nodes keep the fresh route information but do not function actively. For the reason of having minimal overhead, these nodes do not generate any periodic HELLO message to their neighbors for updating their routing tables. Being a gray node does not initialize its own data transfer. However, these nodes do reply to HELLO messages and for packet forwarding these nodes read the route information in the packet header to update their own routing tables. These nodes can become white nodes if they do not communicate for a specified period of time by assigning zero value to KAB. If a gray node intends to transfer or is requested to receive data packets, it becomes a black node by increasing the value of KAB by one.
- (c) *Black Nodes*. These nodes actively transmit data over the network to different targets. These operate on the fundamental approaches of DSR and OLSR protocols. If a node does not transmit data for a defined period of time, it becomes a gray one by decrementing the value of its KAB by one. On each new activity of data transfer or on receiving data packets, these increment the value of their KABs by two to present their increased activity level.

3.3. Nodes' Behavior and Their States Transition. The nodes change their states with the time and their operations. Nodes perform different tasks during their varying states. Moreover nodes behave differently in various cases. The scenarios are explained in Table 1.

- (a) *Packet Forwarding Nodes*. White nodes change their state to gray nodes while gray and black nodes keep their same states but increase the KAB to present their frequency of activeness.
- (b) *Data Transfer Initialization*. White and gray nodes are converted into black to transfer data. However, black nodes increase the value of their KAB to indicate their level in the same state.
- (c) *KAC Clock*. The KAC for white node is always deactivated to avoid any extra resource usage. It decrements values of KAB in gray and black nodes by 2 and 1, respectively.
- (d) *HELLO Messages*. For sensing the presence of neighboring nodes, HELLO messages are generated in which white nodes do not participate. Gray node only cooperate to HELLO messages but do not generate.

TABLE 1: Nodes' behavior in different situations.

Situation	White	Gray	Black
Packet forwarding	(i) Increment value of KAB by 1 (ii) Activate KAC (iii) Update its own routing table (iv) Forward packet	(i) Increment value of KAB by 2 (ii) Reset KAC (iii) Apply route shortening procedure (iv) Forward packet	Increment value of KAB by 2 (i) Reset KAC (ii) Apply route shortening procedure (iii) Forward packet
Data transfer initialization	(i) Increment value of KAB by 2 (ii) Activate KAC (iii) Initiate route discovery	(i) Increment value of KAB by 1 (ii) Reset KAC (iii) Use routing table. If route is not available, then initiate route discovery	(i) Increment value of KAB by 2 (ii) Reset KAC (iii) Use routing table. If route is not available, then initiate route discovery
KAC clock	[DISABLED]	(i) Decrement value of KAB by 2 (ii) If value of $KAB \leq 0$, then (a) $KAB = 0$, (b) deactivate KAC, and (c) clear all queues (iii) Remove older NSA entries	(i) Decrement value of KAB by 1 (ii) Reset KAC (iii) Remove older NSA entries
HELLO messages	(i) Do not reply and do not generate	(i) Reply but do not generate (ii) Alter NSA	(i) Reply and generate (ii) Alter NSA

3.4. Route Discovery. Inspired from DSR protocols, route discovery provides the source route to a node A which decides to send a packet to destination F. It is only used when node A, the initiator, does not know the route information to transfer data packets to the target node F. In most of the cases, once the network is established, white nodes need a route discovery procedure.

When a node does not know about the route to the desired destination, it initiates a route discovery by flooding a RREQ to all its neighbors. Upon receiving the RREQ, the target node returns the source node to a RREP to establish a route. A target nonblack node also changes its state to the black node. The target node discards the repeated same RREQ to avoid any replication in the transmission of control packets. Upon reception of RREP the source node stores the route information in its routing table for transmission of subsequent packets to the target. For sending RREP the target node uses its own route information primarily and uses the reverse route secondarily. White nodes reply on the same route of RREQ packet. For handling nonavailability or void result of route discovery the source node also keeps a copy RREQ in its Route Request Queue.

To reduce the overhead of excessive route discovery packets the intermediate nodes can provide their own route information stored in their routing tables. Moreover the intermediate nodes can direct the RREQ to a specific route to avoid unnecessary flooding of control packets in the network. For example, in Figure 2, node A initiates a RREQ by flooding it to nodes B, C, and D. Nodes B and C know the route; therefore the route discovery is directed on a specified route through F. Being white node, node D does not know about the route and forwards the RREQ to all of its neighbors. F forwards the RREQ with a given route only and discard the packets with blank route information.

3.5. Route Maintenance. When a node transfers data on a known route, the intermediate nodes update their routing tables by reading the address information in the data packet

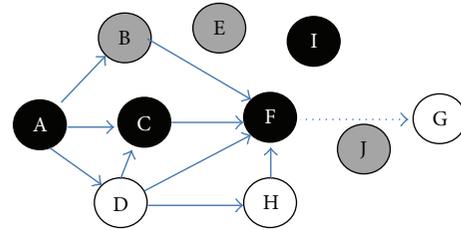


FIGURE 2: Route discovery: reducing RREQ flooding.

header. A forward white node updates its routing table by reading the routing information in the packet header and upgrade itself to a gray node. A source node may find multiple gray and/or black nodes towards the destination. It always selects an intermediate neighbor with lowest value of KAB for data transfer towards the destination node with an intension for reducing load on highly active nodes.

Additionally, the black nodes proactively maintain the route information in their routing tables by the mechanism of sensing their neighbors. Each black node periodically generates a HELLO message to its neighbors. White nodes, not participating in this practice, do not return any reply to HELLO messages. Gray nodes do not generate any periodic message for route maintenance having the purpose to reserve their limited resources available for future data transfers. However, they cooperate with black nodes demonstrating their presence by replying the HELLO messages generated by black nodes. If a black node does not receive the reply of HELLO message, the route is considered as broken.

In most of the cases of data or control packets transfer through intermediate nodes, the gray nodes forward the packets according to the route described in the packet header. In some situations the route shortening procedure changes the original routes. In case the onward route from an intermediate node is broken then the intermediate node floods the packet to all of its neighbors to search out the route. Take example in Figure 3, in Case 1, node A has a route

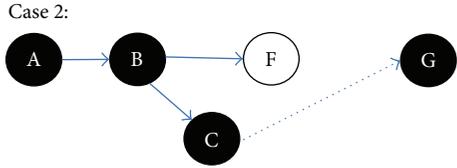
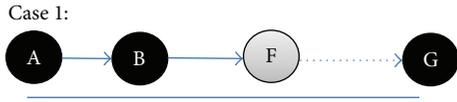


FIGURE 3: Route maintenance.

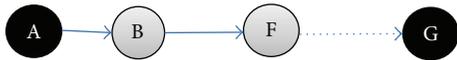


FIGURE 4: Route maintenance.

to node G through nodes B, F and some other intermediate nodes. B senses the presence of node F by receiving replies to its periodically generated HELLO messages. Later, in Case 2, node A wants to transmit data to G through its previously stored route. But in this case node F has changed its state to white and does not reply to B's HELLOs. Now node B detects that the route is broken or suspicious. To avoid an error message for node A, node B piggybacks Route Modification Request (RMR) with the same data packet and floods it to all its neighbors to check the alternate or lost route. Node C knows the route to G; by receiving such packet with destination G, it modifies the route with its own and forwards the packet. Additionally node C also acknowledges the source node A with the new route by sending a control packet back to A through B. Moreover node B is also benefited from new route information by reading the generated packets from node C to node A. In such type of route recovery the nodes usually use hop count to reduce congestion load on the links during flooding.

In Figure 4 two simultaneous gray nodes are connected in a route. It is sure that B does not update the presence status of link with F. On receiving a packet node B forwards it according to its previously updated routing table entry matched with the route information in the packet. Due to gray states of connected nodes, there is possibility of link breakage at any intermediate node. The node by knowing the error in link does the same procedure as explained in Figure 3. In case of failure the initiator ultimately receives the link error generated by the node knowing the route breakage.

Consider the situation in Figure 5, node F does not keep any routing information. It straightaway forwards the packet according to the provided route. If the route does not exist, then it floods the packet to its neighbors adding an RMR with the packet. On success node F converts to a gray node by updating its route information and in case of failure the source node A is acknowledged with a route error.

In some cases, like in Figure 6, node B does not find the route to node G; even after flooding a packet, it returns node A to a route error. Figure 6 shows the inaccessibility of node G, either disappeared or broken route, through node B from node A.

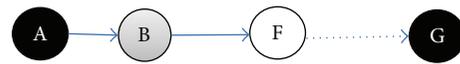


FIGURE 5: Route maintenance.

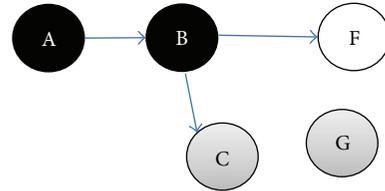


FIGURE 6: Route maintenance.

3.6. Route Shortening Procedure. A forward gray or black node, before forwarding the RREQ packet, applies the route shortening procedure to take or give advantage of the route information. A gray or black forward node compares the provided route information with its stored one and automatically shortens the route by replacing the given route with stored route in its own routing table, if it detects its own route to be shorter than the provided one. If the provided route is shorter, then it updates its own routing table with the newer route. Node B has alternate routes to node G in Figure 7. Due to mobility or suspension of nodes, the routes may break with the passage of time. If a data or control packet is replied with a route error due to broken or nonavailable link, all the forward nodes in the network, except white, also update their routing tables accordingly. White nodes do not keep fresh lists of routes in their routing tables. Therefore these nodes do not help in any sort of additional routing operations.

The duplication of nodes in a route is eliminated to shorten the route. Like in Figure 8 node F is repeated. F can be deleted from the route accordingly.

4. Simulation Results

The work has been simulated by using NS2.35 under Ubuntu operating system. Table 2 shows the parameters used in the simulation.

Figure 9 shows the delay comparison of SALMA with other routing protocols. In case of DSR the values of end-to-end delays suddenly change upward at different time intervals. This is due to its proactive route discovery method. The normalized end-to-end delays with various pause times is examined and it is found that SALMA has less average delay times when compared to the pure proactive and reactive protocols, that is, DSR and OLSR. SALMA is stable just like ZRP and HOPNET due to their hybrid nature. At start SALMA and DSR both give higher values, but with the passage of time and updates in routing table, SALMA starts proactive routing and gives lesser delays like OLSR.

OLSR, according to Figure 10, gives better results than the other studied protocols. As the number of nodes is increased the end-to-end delays in OLSR and SALMA are decreased. SALMA, like other hybrid protocols HOPNET and ZRP, gives reasonably acceptable results as compared to DSR. It gives

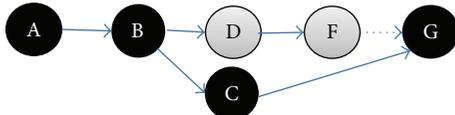


FIGURE 7: Route shortening procedure.

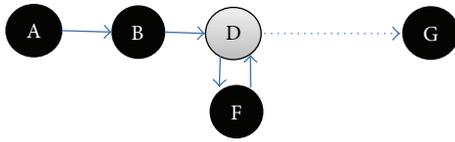


FIGURE 8: Repeated node.

TABLE 2: Simulation parameters.

Parameter	Value
Network size	1 km × 1 km
CBR data rate	1500 kbps
CBR packet size	64 bytes
Interface queue type	Priority queue
Queue length	40 packets
Physical mode DSSS rate	11 Mbps
Node speed	Random
Node pause time	Random
Node transmission power	7.5 dBm
Number of nodes	Up to 200
Protocols test	SALMA, DSR, OLSR, ZRP, HOPNET, LEACH, and PEGASIS

less delays with higher number of nodes just like OLSR. Moreover, SALMA has very minor difference with OLSR for higher number of nodes too. As the number of nodes is increased, the number of zones is also increased. ZRP and HOPNET protocols produce higher delays due to their increased reactive route maintenance among higher number of zones.

In Figures 11 and 12, we can see that SALMA gives moderate value of overhead. DSR presents lower routing load as compared to others. However, SALMA beside other positive results also gives reasonably acceptable values in this test. Due to proactive nature of OLSR, the ratio of growth in routing load is very high with increased number of nodes.

Figure 13 shows the comparison results of SALMA with HOPNET and ZRP for routing overhead. In HOPNET and ZRP, the periodic control packets within the zone are generated to maintain the routes. While route discoveries are performed among the zones reactively. SALMA is a state-based protocol and does not rely on the topological and zonal distribution of nodes. It gives a space for energy efficacy and keeps some nodes asleep due to which the protocol adopts both proactive and reactive algorithms by considering the states of nodes. The results show that SALMA produces lower overhead with different number of nodes as compared with the other two pure hybrid protocols. The main reason

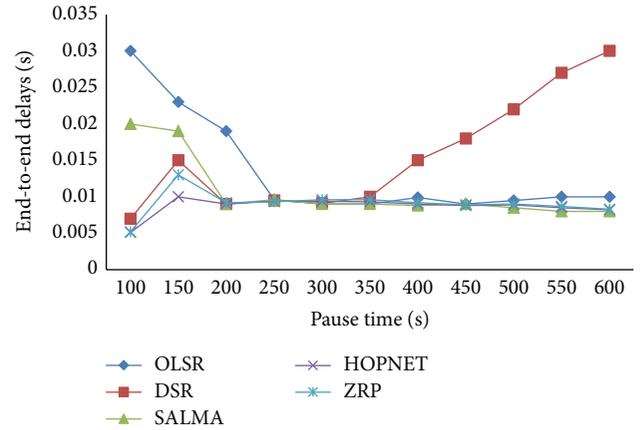


FIGURE 9: End-to-end delays versus pause time.

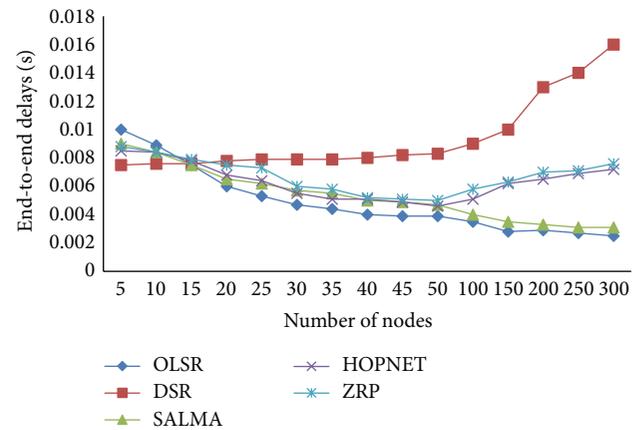


FIGURE 10: End-to-end delays versus number of nodes.

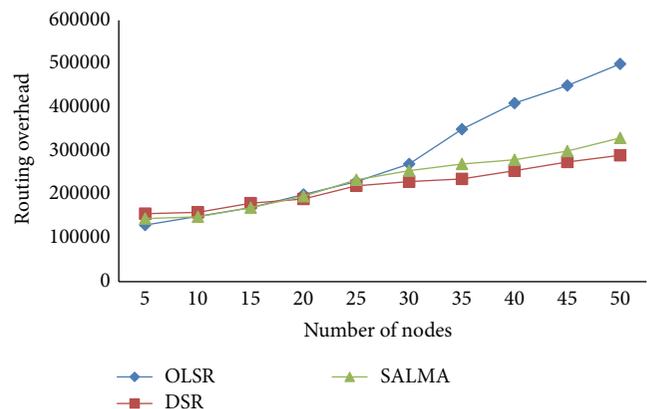


FIGURE 11: Routing overhead versus number of nodes.

is that the typical hybrid protocols generate many control messages without considering the states of nodes. With the increased number of nodes the number of control messages is also increased, which in turn tends to increase the routing overhead.

The graph in Figure 14 shows the energy consumption of nodes by DSR, OLSR, ZRP, HOPNET, and SALMA with

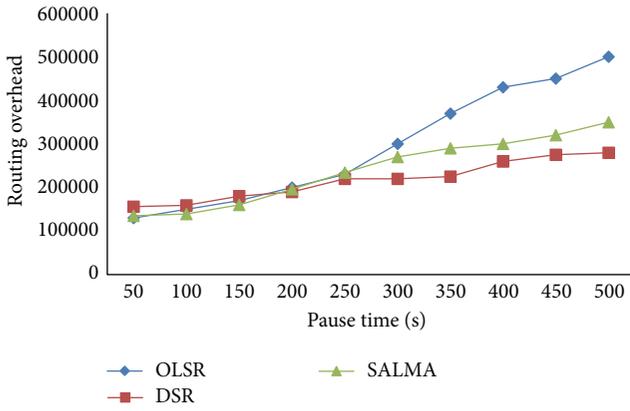


FIGURE 12: Routing overhead versus pause time.

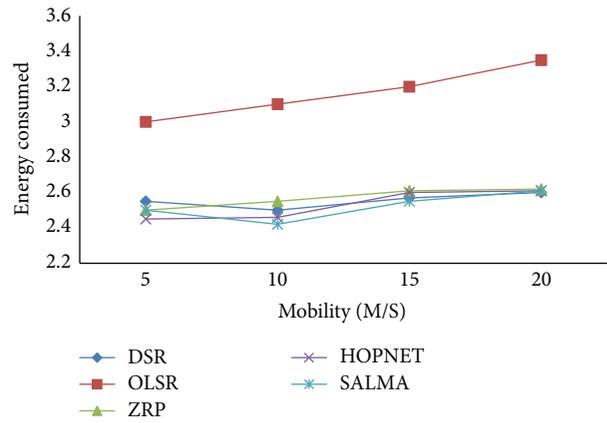


FIGURE 14: Energy consumption with nodes' mobility.

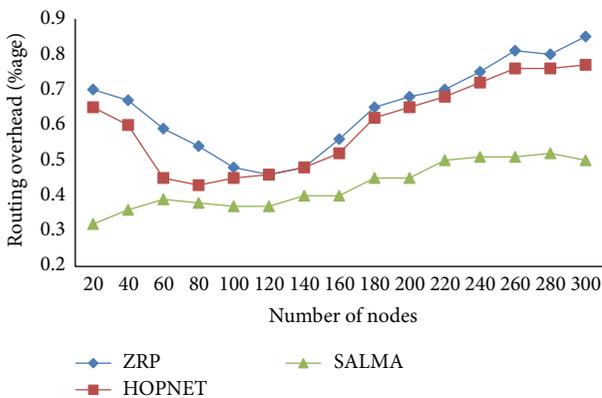


FIGURE 13: Routing overhead compared with HOPNET and ZRP.

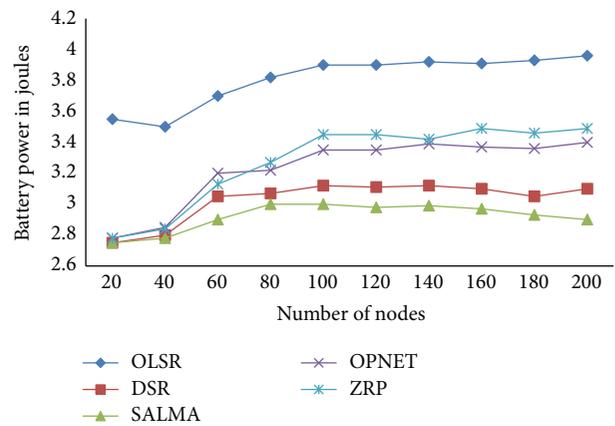


FIGURE 15: Average of nodes' consumed power.

different mobility rates when applied in a WSN. Due to the hybrid nature of ZRP and HOPNET, both give better results than OLSR. SALMA also produces similar and somehow better results than ZRP and HOPNET. It is because this protocol does not use zones and utilize maximum benefit of combining the proactive and reactive algorithms. SALMA protocol does not put routing burden on nontransmitting nodes to reduce their energy consumption.

Figure 15 shows that, with SALMA protocol, nodes consume less energy than other compared protocols. The state-aware mechanism in SALMA protocol prohibits the idle nodes to not consume their power on periodic route maintenance procedure. Therefore, as the number of nodes increases, the average energy consumption reduces. OLSR consumes relatively very high power due to its proactive nature of route maintenance. The energy consumption by ZRP and HOPNET is also increasing with enlarged number of nodes.

Figure 16 shows that PEGASIS produces better results than LEACH, which is up to 300%, with different death rates of nodes at various recorded rounds. SALMA for being noncluster and using three states of nodes functionality outperforms the both in this experiment. It gives much better results than PEGASIS and LEACH for having the least number of died nodes at different rounds.

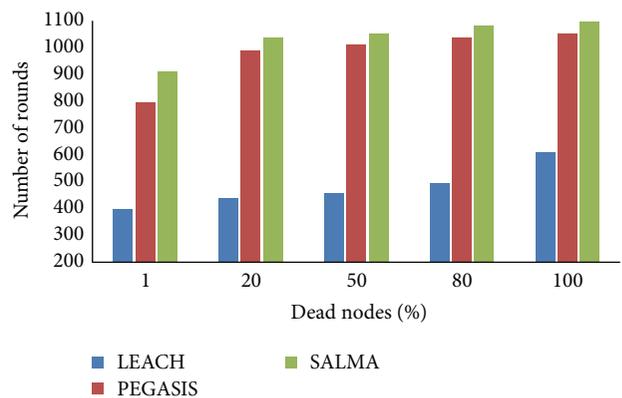


FIGURE 16: Performance results compared with cluster based protocols.

5. Conclusion and Future Work

SALMA protocol is proposed to handle the problems related to proactive and reactive protocols by merging them into one. The work used best of DSR and OLSR protocols into one. Moreover, SALMA introduced the three states of nodes, which in turn help the network to be more energy efficient.

The nonactive nodes are kept in sleeping mode so that they can save more power and extend their lives. Some new data structures are also introduced to support the state-aware mechanism in the protocol. SALMA uses DSR protocols for initial discovery of routes. It uses OLSR protocol for maintenance of routes to avoid delayed data communication due to reactive rediscovery of routes. Results show that SALMA gave moderate and aggregately better results as compared to DSR, OLSR, ZRP, and HOPNET protocols.

The work can be further extended by consideration of some other parameters like mobility speed of nodes, nodes' transmission power, security issues, and positioning of nodes. Features of some other standard reactive and proactive protocols can also be implemented in the current system to attain more efficient functionalities.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

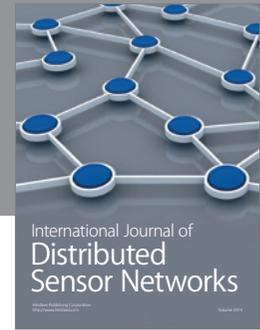
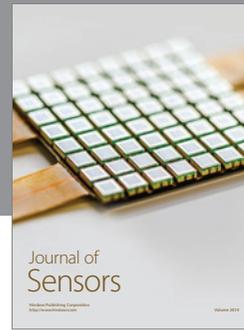
Acknowledgment

The authors would like to extend their sincere appreciation to the Deanship of Scientific Research at King Saud University for funding this Research Group no. RG-1435- 037.

References

- [1] A. G. A. Elrahim, H. A. Elsayed, S. E. Ramly, and M. I. Magdy, "An energy aware WSN geographic routing protocol," *Universal Journal of Computer Science and Engineering Technology*, vol. 1, no. 2, pp. 105–111, 2010.
- [2] S. Sharma, R. K. Bansal, and S. Bansal, "Issues and challenges in wireless sensor networks," in *Proceedings of the IEEE International Conference on Machine Intelligence Research and Advancement (ICMIRA '13)*, pp. 58–62, Katra, India, December 2013.
- [3] S. Gupta and K. C. Roy, "Comparison of different energy minimization techniques in wireless sensor network," *International Journal of Computer Applications*, vol. 75, no. 18, pp. 20–26, 2013.
- [4] R. Kaur and M. K. Rai, "A novel review on routing protocols in MANETs," *Undergraduate Academic Research Journal*, vol. 1, no. 1, pp. 103–108, 2012.
- [5] L. J. G. Villalba, J. G. Matesanz, A. L. S. Orozco, and J. D. M. Díaz, "Auto-configuration protocols in mobile ad hoc networks," *Sensors*, vol. 11, no. 4, pp. 3652–3666, 2011.
- [6] S. Dhenakaran and A. Parvathavarthini, "An overview of routing protocols in mobile ad-hoc network," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 2, pp. 251–259, 2013.
- [7] K. Tripathi, M. Pandey, and S. Verma, "Comparison of reactive and proactive routing protocols for different mobility conditions in WSN," in *Proceedings of the International Conference on Communication, Computing and Security (ICCCS '11)*, pp. 156–161, Odisha, India, February 2011.
- [8] P. Shahbazi, "A survey on hybrid routing protocols in MANETS," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 1, no. 12, pp. 882–887, 2013.
- [9] Z. J. Haas, M. R. Pearlman, and P. Samar, *Internet Draft—Mobile Ad Hoc Networking (MANET)*, Working Group of the Internet Engineering Task Force (IETF), Fremont, Calif, USA, 1997.
- [10] V. Ramasubramanian, Z. J. Haas, and E. G. Sirer, "SHARP: a hybrid adaptive routing protocol for mobile ad hoc networks," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc '03)*, pp. 303–314, ACM, Annapolis, Md, USA, June 2003.
- [11] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," RFC 3626, 2003.
- [12] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, pp. 153–181, Springer, New York, NY, USA, 1996.
- [13] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, 2004.
- [14] D. Bhattacharyya, T.-H. Kim, and S. Pal, "A comparative study of wireless sensor networks and their routing protocols," *Sensors*, vol. 10, no. 12, pp. 10506–10523, 2010.
- [15] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, 2002.
- [16] J. Wang, E. Osagie, P. Thulasiraman, and R. K. Thulasiram, "HOPNET: a hybrid ant colony optimization routing algorithm for mobile ad hoc network," *Ad Hoc Networks*, vol. 7, no. 4, pp. 690–705, 2009.
- [17] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.
- [18] A. M. Okazaki and A. A. Frohlich, "AD-ZRP: ant-based routing algorithm for dynamic wireless sensor networks," in *Proceedings of the 18th International Conference on Telecommunications (ICT '11)*, pp. 15–20, IEEE, Ayia Napa, Cyprus, May 2011.
- [19] A. M. Okazaki and A. A. Frohlich, "Adapting HOPNET algorithm for wireless sensor networks," *IEEE Latin America Transactions*, vol. 8, no. 4, pp. 30–35, 2009, Proceedings of the 9th International Information and Telecommunication Technologies Symposium.
- [20] L. J. G. Villalba, A. L. S. Orozco, A. T. Cabrera, and C. J. B. Abbas, "Routing protocols in wireless sensor networks," *Sensors*, vol. 9, no. 11, pp. 8399–8421, 2009.
- [21] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks," *Ad Hoc Networks*, vol. 2, no. 1, pp. 1–22, 2004.
- [22] S. Maruthamuthu and S. Sankaralingam, "Hybrid hop count based multiple Qos constraints routing protocol with mobility prediction for manet," *European Journal of Scientific Research*, vol. 52, no. 4, pp. 604–617, 2011.
- [23] A. Al-Rokabi and C. Politis, "SOAP: a cognitive hybrid routing protocol for mobile ad-hoc networks," in *Proceedings of the 9th International Conference on Cognitive Radio Oriented Wireless Networks (CROWNCOM '14)*, pp. 353–359, June 2014.
- [24] G. Liu, P. Chen, and Z. Hu, "A hybrid routing protocol based on energy-aware in WSNs," *Transducer and Microsystem Technologies*, vol. 6, pp. 25–28, 2012.
- [25] R. Lent, "Simulating the power consumption of computer networks," in *Proceedings of the 15th IEEE International Workshop on Computer Aided Modeling, Analysis and Design of Communication Links and Networks (CAMAD 2010)*, pp. 96–100, Miami, Fla, USA, December 2010.

- [26] F. De Rango, M. Fotino, and S. Marano, "EE-OLSR: Energy efficient OLSR routing protocol for mobile ad-hoc networks," in *Proceedings of the IEEE Military Communications Conference (MILCOM '08)*, pp. 1–7, IEEE, San Diego, Calif, USA, November 2008.
- [27] M. Dhivya and M. Sundarambal, "Hybrid AODV protocol for energy efficient routing in wireless sensor networks," *International Journal of Mobile Network Design and Innovation*, vol. 4, no. 2, pp. 59–64, 2012.
- [28] R. Devi, "QOS enhanced hybrid multipath routing protocol for mobile adhoc networks," *International Journal of Distributed and Parallel Systems*, vol. 3, no. 6, pp. 89–105, 2012.
- [29] Z. Xu, K. Wang, and L. Qi, "Adaptive threshold routing algorithm with load-balancing for ad hoc networks," in *Proceedings of the International Symposium on Web Information Systems and Applications (WISA '09)*, pp. 351–354, Nanchang, China, May 2009.
- [30] A. E. Tümer and M. Gündüz, "Energy-efficient and fast data gathering protocols for indoor wireless sensor networks," *Sensors*, vol. 10, no. 9, pp. 8054–8069, 2010.
- [31] S. Lindsey, C. Raghavendra, and K. M. Sivalingam, "Data gathering algorithms in sensor networks using energy metrics," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 9, pp. 924–935, 2002.
- [32] Z. Mohammad, M. Amir, and M. K. Leili, "Novel hierarchical routing protocol in wireless sensor network," *Procedia Computer Science*, vol. 3, pp. 292–300, 2011.
- [33] J.-S. Chen, Z.-W. Hong, N.-C. Wang, and S.-H. Jhuang, "Efficient cluster head selection methods for wireless sensor networks," *Journal of Networks*, vol. 5, no. 8, pp. 964–970, 2010.
- [34] H. Gou and Y. Yoo, "An energy balancing LEACH algorithm for wireless sensor networks," in *Proceedings of the 7th International Conference on Information Technology: New Generations (ITNG '10)*, pp. 822–827, Las Vegas, Nev, USA, April 2010.
- [35] S. Radhakrishnan, G. Racherla, C. N. Sekharan, N. S. V. Rao, and S. G. Batsell, "DST-a routing protocol for ad hoc networks using distributed spanning trees," in *Proceedings of the 1st IEEE Annual Wireless Communications and Networking Conference (WCNC '99)*, Cat. No.99TH8466, pp. 1543–1547, New Orleans, La, USA, September 1999.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

