Advances in Networks
2016; 4(2): 21-33
http://www.sciencepublishinggroup.com/j/net
doi: 10.11648/j.net.20160402.12
ISSN: 2326-9766 (Print); ISSN: 2326-9782 (Online)

Balanced Multicast Routing Performance in Autonomous Sensor Networks

Abdelrahman S. Halawa, Salah M. Abdel-Mageid

Systems and Computers Department, Faculty of Engineering, Al-Azhar University, Cairo, Egypt

Email address:
abdelrahmanhalawa@gmail.com (A. S. Halawa), smageid@azhar.edu.eg (S. M. Abdel-Mageid)

To cite this article:
Abdelrahman S. Halawa, Salah M. Abdel-Mageid. Balanced Multicast Routing Performance in Autonomous Sensor Networks. *Advances in Networks.* Vol. 4, No. 2, 2016, pp. 21-33. doi: 10.11648/j.net.20160402.12

Received: October 11, 2016; **Accepted**: November 2, 2016; **Published**: December 21, 2016

Abstract: Wireless sensor networks (WSNs) are almost autonomous and dynamic network consisting of a number of wireless nodes with limited energy. Accordingly, multicast routing protocols for WSNs encounter many challenges such as energy limitation, limited bandwidth, and self-configuration when producing multi-hop routes. Most existing multicast routing protocols are designed for ad hoc networks which maximize network throughput regardless energy consumption. However, multicast routing protocols for WSNs must consider the trade-off between minimizing energy consumption and maximizing network throughput. In this paper, balancing multicast routing metrics is considered for WSNs in which the route selection is based on nodes' remaining energy and hop count metrics. In addition, the advisor node of a multicast group is periodically changed every predefined time interval to achieve more energy balancing., The performance of the proposed scheme is evaluated and compared with the existing multicast routing protocols using NS2 in terms of network lifetime, network throughput, packet delivery ratio, end-to-end delay and network control overhead which the impact of traffic load, node density, noise level, and mobility models are taken into account. The results show that the performance of the proposed scheme is better than the performance of existing schemes.

Keywords: Wireless Sensor Networks, Multicast Routing, Balanced Metrics, Node Energy, Hop Count, Network Lifetime

1. Introduction

Autonomous sensor networks play an important role in many applications such as battlefield and disaster relief [1]. Some nodes are responsible for making a decision without assistance from the centralized node (i.e. base station). Since bandwidth, battery energy and mobility are the key issues in such types of applications, a multicast routing has become important which provides a higher packet delivery ratio, less control overhead, less bandwidth usage and less network delay. Moreover, multicast routing reduces the communication cost (i.e., source node sends similar packets to many destinations instead of individually sending to each one) [2], [3], [4] and [5].

Most existing multicast routing protocols have been proposed for MANETs and can be categorized into two types, mesh-based and tree-based. Mesh-based multicast protocols construct a mesh structure instead of a tree structure to connect source node to destination node [2], [6], [7], and [8]. Mesh structure can efficiently deal with link failures due to redundant paths. However, more excessive overhead is performed due to flooding messages.

In contrast, tree-based multicast routing protocols, group members are connected in a tree structure. Such technique guarantees individual path between the source and the destination nodes [9]. There are two kinds of tree-based schemes, source-based tree and shared tree. In the source-based tree, each source node constructs a transmission tree and makes itself as a root [10], [11]. Source-based trees are inappropriate for multipoint-to-multipoint multicast and have troubles with network scaling, due to the periodic flooding necessary to detect new nodes. In contrary, shared multicast tree approaches such as [4], [5] and [12] uses a single tree rooted by a core. In the shared-based tree, the sender unicasts its data to towards the core of the group, then the core (or one of the tree members) sends a data packet to receivers.

The most challenging in WSNs is energy limitation. The mentioned multicast protocols require some adaptation to be appropriate for WSNs. In this paper, the proposed scheme maximizes the network lifetime by balancing the remaining energy of nodes and hop count in multicast route construction (it is called Balanced Multicast Routing, BAMUR). BAMUR scheme calculates Route Score (RS) of all candidate routes using the remaining energy of intermediate nodes and hops count. The route of highest RS is chosen as a transmission route. In addition, we set a predefined energy level to change the advisor of the multicast group to overcome the power exhaustion of group advisor. BAMUR scheme performance is evaluated using the NS2 simulator. A comparative study is introduced to show that BAMUR solution is better than the original MAODV and PDTMRP in network lifetime without affecting network throughput and data packet loss.

The rest of the paper is structured as follows. Section 2 briefly explains the existing works of tree-based multicast routing in MANETs. Section 3 describes the proposed scheme and the life cycle of its messages used in WSNs to show how it overcomes energy limitations. The performance evaluation and simulation results are presented in Section 4. Finally, Section 5 concludes the paper.

2. Related Work

There are many tree-based multicast routing protocols appeared in last decades [4], [12], [13], [14]. The work in [4] extends AODV [15] by adding the multicast feature for MANET, it is called MAODV which keeps a shared tree for each multicast group. Hello messages are periodically broadcast to maintain the group tree and group dissemination. One of the group members is chosen to be a group leader, which is initially the first node needs to join the tree. The route between the source and destination is established by deliberation among network's nodes via three types of messages, route query, route report and multicast startup.

The source node broadcasts a route query packet when it wants to join a multicast group or to send data. An intermediate node forwards such packets to its neighbors. Afterwards, the multicast tree nodes only unicast route report messages to the source node. When the source node receives report messages, it selects the route with the largest sequence number and smallest hops count as a communication route and unicasts a multicast start-up message to enable that route. MAODV is easy to implement; however, it suffers from unbalancing power exhaustion and partitioning problems when it is applied for WSNs.

AMRIS [13] consists of two phases initialization and maintenance. In the initialization phase, a multicast session is established and the best routing path is determined. The source node generates new session message includes a multicast session member ID and routing table. Such message is broadcasted to other nodes. The smallest multicast session member ID is assigned to the source node which waits for a query to join from the receivers. When an intermediate node receives a new session message, it determines its own ID which is greater than parent ID and broadcasts the new session message with its ID. When a new session message reaches a receiver node, it determines its own ID and sends a query to join to parent with the smallest ID. An intermediate node forwards the query to the source node which sends an acknowledgment to receivers. When a receiver receives an acknowledgment, the tree is established. In the maintenance phase, the network topology is maintained. However, tree nodes periodically send beacons in order to show their existence. Consequently, bandwidth is wasted and also many packets are lost as a result of collisions between beacons.

In ADMR [14], active nodes between senders and receivers are only situated by establishing a multicast state animatedly to pledge the shortest delay path between senders and receivers. However, some nodes may be active for a long time to achieve the shortest delay. Consequently, it is unsuitable to be applied for sensor nodes because depleted energy nodes lead to network partitioning. The work in [12] solves the problem of a single node failure and high mobility by creating redundant links between a pair of nodes. The number of redundant links is controlled by the network traffic and mobility. However, establishing a tree with multiple links requires high communication cost which is not appropriate for WSNs.

In addition, many multicast routing protocols [16], [17], [18], and [2] provide an efficient tree maintenance algorithm to fix the route failure due to battery exhaustion of tree member nodes to increase the lifetime. However, the group advisor has not considered in maintenance process, although the group advisor is the first node confronting the battery exhaustion problem. This may lead to split the network group to disjoint groups. On the other hand, the works in [19] [19], [20] solve the partitioning problem by selecting a new tree member node to be the group advisor of the tree. Nevertheless, when selecting the route between source and receiver, it is not taken into account the energy power of the route nodes. The increase in control message overhead leads to a battery exhaustion problem with route nodes.

PDTMRP [5] is a dual-tree-based protocol which divides the transmission data packets into two portions and sends each portion over a different tree. PDTMRP randomly classifies the network nodes into two types group-0 and group-1. The dual-tree is constructed for data packets transmission by building tree-0 for group-0 and tree-1 for group-1. In this work, each node retains two routing tables the routing table and the neighboring table. Each node periodically announcements hello message to determine its neighbors. The source node broadcasts the route query message whenever it has data to send to destination nodes. As soon as a destination node catches an appropriate path, it reverses a route report packet back to the source node; therefore, the routing table is built. This work used a pre-defined energy threshold value to decide whether it needs to rebroadcast the route query. Accordingly, the energy consumption is reduced by using dual trees for transmission. However, the network still suffers from a single node failure problem because a group advisor energy can be rapidly depleted in WSNs. The proposed solution, BAMUR, introduces a balanced multicast routing strategy that overcomes the energy problems as much as possible, and balances the energy consumption among nodes, predicts and solves the partitioning problem.

3. The Proposed Multicast Routing Scheme for WSNs

BAMUR scheme is mainly proposed to improve the performance of multicast routing in WSNs. It contains two stages: *foundation*, and *monitoring*. The foundation stage is proposed to create a multicast group and dynamically choose the best route, for a source node that wants to join or send data to this group, according to new balanced routing metric (it is called route score, RS) which is mainly designed to prolong the lifetime of WSN. On the other hand, the monitoring stage is proposed to help BAMUR scheme to continue working in an efficient manner when BAMUR is forced to change the group advisor node in case of its remaining energy reach to a predefined energy threshold. This stage also prevents the partitioning problem of the multicast group in WSNs achieving more reduction in the number of control messages in BAMUR scheme. In what follows BAMUR stages are illustrated in more details.

3.1. Foundation Stage

As shown above, such stage mainly chooses the best route that achieves balanced routing metrics. Since BAMUR scheme exploits the on-demand routing principles, three types of control messages are adapted to work on BAMUR scheme: query, report and startup. In addition, announcement control message is also required in multicast processing. When a node desires to join or send data to a multicast group, it broadcasts a *query* control message, then a multicast tree member (includes group advisor) unicasts a *report* control message back to the source node (the owner of the *query* message), then the source node unicasts a *startup* message towards the owner node of *report* message to permit this route for multicasting.

This work proposes a multicast routing mechanism for autonomous sensor nodes. Therefore, the role of each node is described in details in BAMUR algorithm. All nodes can be classified into three categories: source, intermediate, and member. The source node is a normal node that wants to join a multicast group, or a group member has data to send. The intermediate node is a non-member node that receives a multicast packet (query, report or startup) and forwards it to a specific neighbor or all neighbors according to the packet type. The member node is a tree member, group member, or a group advisor of the current multicast group. BAMUR foundation stage is illustrated in Fig. 1.

As shown in Fig. 1, three control messages (query, report, and startup) are used among nodes. Initially, when a source node wants to join a multicast group, it broadcasts a query message, indicated by arrows (1) and (2), to a multicast member node via intermediate nodes (non-member nodes).

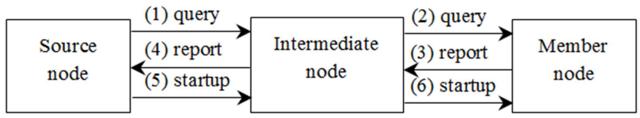
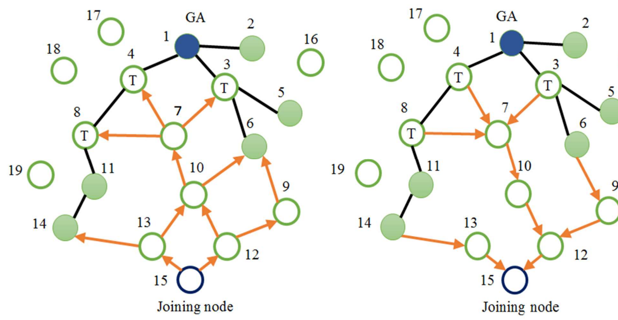


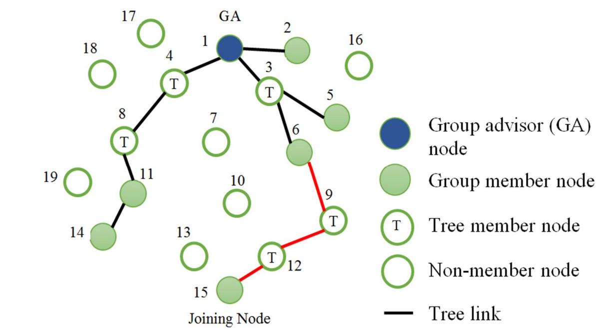
Fig. 1. *BAMUR foundation stage.*

When a member node (tree/group member or group advisor) receives a query message, it broadcasts a report message, indicated by arrows (3) and (4), to the source node via intermediate nodes. Finally, the source node chooses the best route to multicast group according to the proposed metrics and sends a unicast message (startup message) to members, indicated by arrows (5) and (6), via intermediate nodes.

In this section, a multicast foundation algorithm for BAMUR scheme is introduced as shown in Table (1). To facilitate understanding BAMUR foundation algorithm, an example of a multicast tree is illustrated in Fig. 2. There are six multicast group members (1, 2, 5, 6, 11 and 14) in which node (1) is a group advisor. There are three tree members (3, 4 and 8) while nodes (7, 9, 10, 12, 13, 15, 16, 17 and 18) are non-member nodes. In what follows, the propagation of control messages during the foundation stage is illustrated.



(a) query message scenario (b) report message scenario



(c) Multicast tree after startup

Fig. 2. *Multicast Establishment.*

3.1.1. Query Message Propagation

When a source node (it is called here joining node) wants to join a multicast group, it prepares a *query* message to broadcast it (i.e., node (15) is a source node in our example). Several fields are required in the prepared *query* message such as join indicator, healing indicator, broadcast ID, source address, source sequence number, destination address, destination sequence number and hop count. The healing indicator (HI) field is used for fixing the multicast group as shown below (Healing Damaged Links section).

Every node in the network keeps two counters, a sequence number, and a broadcast ID. The sequence number confirms that the routes to the current node are fresh while the broadcast ID and source node IP address are used as a key to identify the query of the source node. Whenever any node obtains new neighbor information, its sequence number is incremented, and the broadcast ID is incremented for each time the source node broadcasts query message. The steps from (S2 to S8) in foundation algorithm, as shown in Table (1), shows how a source node (15) prepares a query message.

Table 1. *An Algorithm for Multicast Foundation in BAMUR scheme.*

	BAMUR Foundation Algorithm
At source node (s)	
S1:	**IF** node (s) wants to join a multicast group **THEN**
	/* node (s) creates a *query packet* (queryPacket) */
S2:	node(s). queryPacket.JI ← true /* JI refers to join indicator */
S3:	node(s). queryPacket.destAddr ← multicastGroupAddr
S4:	node(s). queryPacket.destSeqNo ← multicastGroupSeqNo
S5:	node(s). queryPacket.broadcastID ← BroadcastID(s)++
S6:	node(s). queryPacket.sourceAddr ← Source_Addr(s)
S7:	node(s). queryPacket.sourceSeqNo ← Source_Seq(s)
S8:	node(s). queryPacket.hopCount ← 0
S9:	Set packetCounter:= 0
S10:	**WHILE** packetCouter < No_of_Trials
S11:	node(s).send(queryPacket(packetCouter))
S12:	node(s).waitUntil (report is received or query timeout);
S13:	**IF** query timeout **THEN**
S14:	packetCouter++
S15:	**ELSE IF** reportPacket (k) is received **THEN**
S16:	node(s).computeRS(reportPacket (k))
	node(s).selectRouteToGroupAdvisor (highest RS)
S17:	/*nodes (s) creates startup packet (startupPacket) and unicasts it towards the group advisor*/
S18:	node(s).nextHop.enabledIndicator ← true
S19:	node(s).unicastStartupMsg ( )
S20:	node(s).startupPacket.destAddr ← multicastGroupAddr
S21:	node(s). startupPacket.sourceAddr ← sourceAddr
S22:	nodes(s).send(startupPacket)
S23:	return;
S24:	**END IF**
S25:	**END WHILE**
S26:	**IF** packetCount >= number-of-trials **THEN**
S27:	set node (s) as a multicast group advisor
S28:	node (s).grpSeqNo ← 1
S29:	**END IF**
S30:	**END IF**
At intermediate node (i)	
I1:	**IF** node (i) already received queryPacket(packetCounter) **THEN**
I2:	node (i) discards queryPacket(packetCounter)
I3:	**ELSE**/* query packet is received for first time */
	/* node (i) updates *query packet* to forward it */
I4:	node (i). updateQueryTable ( )
I5:	node (i).updateMulticastRouteTable ( )
I6:	node (i).nextHop ← qp(j).sourceAddr
I7:	node (i).seqNo ← seqNo(i) ++
I8:	node (i). forwardToNeighbors(qp(j))
I9:	**END IF**
I10:	**END IF**
I11:	**IF** node (i) received reportPacket **THEN**
I12:	node(i). reportPacket. mGrpHop ← node(r). reportPacket.mGroupHop++
I13:	node(i). reportPacket.energyList.add(node(i).energy)
I14:	node(i).updateMulticastTable ( )
I15:	node(i).forwardReportMsg (rp(j))
I16:	**END IF**
I17:	**IF** node (i) received startupPacket **THEN**
I18:	node (i) enables the best next hop
I19:	node (i).forwardStartupMsg (next Hop)
I20:	**END IF**
At group/tree member or group advisor (g)	
G1:	**IF** node (g) received queryPacket(packetCounter) **THEN**
	/* node (g) creates report packet and unicasts it towards the source node*/
G2:	node(g). reportPacket.destAddr ← queryPacket (packetCount).destAddr
G3:	node(g). reportPacket.destSeqNo ← multicastGroupSeqNo
G4:	node(g). reportPacket.mGrpHop ← 0
G5:	node(g). reportPacket.energyList.add (node(g).energy)
G6:	node(g). reportPacket.hopCnt ← node (g).distanceTo (groupAdvisor)
G7:	node(g).groupAdvisorAddr ← multicastGroupAdvisorAddr
G8:	node(g).updateMulticastRouteTable ( )
G9:	node(g).send(reportPacket)
G10:	**END IF**

The source node sends a *route query message* when it needs to associate with a multicast group, or when it has data to send to a multicast group and it does not have a renewed route to that multicast group in its routing table. When a source node needs to join the multicast group as illustrated in our example, it sets join indicator, as shown in Line (S2), of the route query; in such case, it broadcasts the route query. On the other hand, when the source node knows a valid route to a node querying a route to that multicast group before, it preserves the indicator unset; in such case, it unicasts the route query according to the information in the routing table.

After sending a *query message*, a source node waits until it receives a report message from a multicast member or query timeout occurs, as shown in Line (S12). When a report is not received in specified time interval, it increments a packet counter as shown in Line (S14) and resends the *query message*. After source node reaches the number of trials, source node sets itself as a group advisor and reset the sequence number of the multicast group to 1, as shown in Lines (S27 and S28), because it assumes that the multicast group is isolated or does not exist.

Nodes (12) and (13) are non-member nodes close to the source nodes. Such nodes are intermediate nodes that receive the query message from the source node. An intermediate node updates its query and multicast tables as shown in Lines (I4-I8); more details about query and multicast tables are introduced in [4]. To avoid flooding the network, an intermediate node check if the query packet is received before, as shown in Line (I1). Nodes (12) and (13) forward the *route query message* to neighbors (9, 10 and 14).

Since nodes (9 and 10) are intermediate nodes, they still forward this query which node (9) forwards to node (6), and node (10) forwards to nodes (6 and 7) as shown in Fig. 2(a). Node (7) forwards the query to nodes (3, 4 and 8). Eventually, there are three tree members (3, 4 and 8) and two group members (6 and 14) receives that query message and get ready to report the source node.

3.1.2. Report Message Propagation

Report message includes the following fields: healing indicator, update indicator, destination sequence number, destination address, multicast group hop, hop count, energy list, and lifetime. When a node is a member of the multicast group's tree and receives a query from the source node, it unicasts a *report message* towards it. For example, the responding nodes (3, 4, 6, 8, and 14), as shown in Fig. 2(b), update its routing table by inserting the next hop node's details information in its routing table and then produce a report and unicast that report back to nodes (7, 9, and 13) which they obtain information about them from the received query message, as shown in Lines (G2 to G9).

The *report message* contains information about the multicast group, such as the sequence number and the IP address of the multicast group advisor. Furthermore, it includes the *mGrpHop* field (multicast group hop count). This field is firstly set to zero and each time the report message is forwarded the multicast group hop count field is incremented. When a node receives a report message, it adds its residual energy to *energy List* field. By the time, the report message will have reached the source node, it will contain the remaining energies of all nodes from the source node to a member node of the multicast tree (the node that unicasts the report message).

When an intermediate node receives a *report message*, it updates or inserts this record in its routing table. The forward path to the source node is created. When the *report message* is received by the source node, the multicast group hop field is updated by the distance (hop count) from the source node to the tree member node that unicasts the report message. The IP address of the group advisor is also placed in a "*Group Advisor Address*" field, see Lines (I12-I15).

3.1.3. Startup Message Propagation

Startup message includes the following fields clip indicator, group advisor indicator, source address, source sequence number, destination address. The clip indicator is used for clipping the tree and group advisor indicator is used for choosing a new group advisor, are described below. Since such step finalizes selecting the best route for the source node. BAMUR scheme introduces a new routing metric called *route score* (RS) to balance node energy in the network and prolong its lifetime. The current energy for node (i) is expressed as follows.

$$\bar{E}_i = h_{avg}\left(\frac{E_i}{E_{init}}\right) \quad (1)$$

Where E_i represents the current energy for node (i) in Joules, E_{init} represents the initial residual node energy. Assuming the average number of hop counts in such network is h_{avg} and $\bar{E}_i$ represents the normalized energy for the current node (i) where i represents its index in energy list. Assume h_{avg} equals 10, a node begins with normalized energy equals 10 and its energy gradually decreases to reach its threshold level (low energy level); for instance, when a node energy reaches one (10% of initial energy), it can be considered a low energy, to avoid it as possible. The proposed metric can be expressed as follows.

$$RS = \frac{\sum_{i=1}^{n} \bar{E}_i^2}{(2h^2+1)\times\left(1+\sum_{j=1}^{n-1}\left(\bar{E}_j-\bar{E}_{j+1}\right)^2\right)} \quad (2)$$

where *n* represents the size of energy list or sorted energy list, $\bar{E}_i$ represents the normalized current energy for node with index (i) in energy list where i ∈{1, 2, ….., n}, $\bar{E}_j$ represents the normalized current energy for node with index (j) in sorted energy list where j ∈{1, 2, ….., n}and h represents the multicast group hop count for this route. The RS metric contains three terms; the first term represents the sum of energy square values for nodes belongs to the current route; $\sum_{i=1}^{n} \bar{E}_i^2$. Obviously, such term must be maximized to prolong the network lifetime. On the other hand, the second term represents the sum of normalized energy difference square for consequent nodes (j) and (j+1) in the sorted energy list; $1+\sum_{j=1}^{N-1}\left(\bar{E}_j-\bar{E}_{j+1}\right)^2$. This term must be minimized to balance the energy consumption in the entire network. Such term is incremented to avoid the sum of normalized energy difference square equals zero. This condition may occur when all nodes in the same route approximately have the same energy. On the other hand, the last term is $(2h^2+1)$ which represents the square value of hop count. It is incremented by one to avoid hop count equals zero. This term must be also minimized to achieve more energy saving, in addition to minimum latency. There is a contradiction between the objective of the first term and the objective of the last terms. Accordingly, RS metric is represented by dividing the first term by the multiplication of the last terms to choose the route that achieves maximum RS.

Those routes are sorted according to their RS values in descending order. The route with maximum RS is chosen constrained by the Lower Node Energy (LNE) in sorted energy list. When such route has LNE less than the threshold level, the route is discarded and BAMUR scheme moves to next route and so on. The probability that there exists a route with maximum RS and LNE less than the threshold value is very small at the beginning of network lifetime. However, such probability gradually increases, especially when most nodes deplete large part of its energy. The worst case, all routes have LNE less than the threshold value. In such case, the route with the minimum hop count is chosen regardless the route remaining energy.

For example, as shown in Fig. 2 (c), once a source node (15) broadcasts a *query message* to a multicast group, it overwhelmingly accepts more than one *report message*. The source node must select only one from such reports as the next hop. According to this way, only one extra branch is added to the multicast tree, and thus the close loop routes are avoided. The source node sends the query and waits a time of period equal to *route detection timeout,* then it selects the route that achieves the highest RS to be its route to the multicast group.

Assuming current normalized node energy is 6.5, 6.8, 6, 7, 7.5, 5.5, 6, 5.8, 6 and 5.3 for nodes 3, 4, 6, 7, 8, 9, 10, 12, 13 and 14, respectively. Therefore, the RS of route <15, 12, 10, 7, 3> is 3.78, RS of route <15, 12, 10, 7, 4> is 4.63, RS of route <15, 12, 10, 7, 8> is 4.1, RS of route<15, 12, 9, 6> is 15.46, and RS of route <15, 13, 14> is 14.53. Accordingly, the route <15, 12, 9, 6> with node energy <5.8, 5.5, 6> is selected (the LNE of this route is greater than the threshold) to be the data transmission route, while it ignores other routes. At the end of the *route detection timeout* period, the source node enables the selected next hop (according to the RS value) in its multicast routing table; afterwards, it unicasts a startup message to the next hop. The destination address field is set to the IP address of the multicast group, as shown in Lines (S16-S23).

When the intermediate node (12) receives the startup message, then it asks its routing table to select the next hop for its route to the multicast group and it enables the matching entry in the table then unicasts a startup message to the next hop, as shown in Lines (I18 and I19). This procedure continues till the node (6) that initiated the report is reached. Other intermediate nodes, that are not involved, remove the entry for the corresponding query node because they do not receive a startup message within a time equal to multicast *tree build time*. Figure 2 (c) clarifies how a multicast tree is constructed. The *startup message* guarantees that the multicast tree has only one path to any tree node. To prevent the probability of data packets being transported to a source node through multiple next hops before a *startup message* is received, intermediate nodes forward data packets only via enabled routes on their multicast route table.

3.1.4. Announcement Message Propagation

The group advisor node remains the advisor of a multicast group until two separate parts of the multicast tree are combined, or till it chooses to leave the multicast group. BAMUR scheme exploits the on-demand routing principles and gives the multicast group advisor the responsibility of conserving the multicast tree. At every *group announce interval* time, group advisor broadcasts a Group Announcement (GAN) message. When a node receives a GAN message, it updates its query table (query table structure is defined in [4]). The GA message is also used for replacing the group advisor and merging two separate parts of multicast trees.

3.2. Monitoring Stage

The monitoring stage is proposed in the BAMUR scheme to replace the group advisor when its energy reaches a predefined energy threshold and to conserve the multicast tree. In what follows, monitoring stage functions are illustrated in details.

3.2.1. Group Advisor Replacement

As mentioned above, the first node starting a multicast group is selected as a group advisor which is responsible for healing the group tree by periodically broadcasting Group-Announcement (GAN) messages in the whole network. The group advisor also keeps the group sequence number, which is propagated on the network through GAN messages. It is expected in WSNs, the multicast group advisor energy is fully exhausted, resulting in an increase in network overhead and loss of data packets.

Unnecessary control messages flooded into the network cause battery power consumption and reduce the network lifetime and overall throughput. BAMUR scheme solves this issue by setting *Replacement Timer* to periodically change the group advisor in a *predefined time interval*. Such interval finishes when all multicast members take their places as group advisors. At this moment, the replacement timer is reset and starts a new interval. In what follows, we show how the group advisor replacement phase works.

Table 2. *Main fields of GAN message with RTC indicator.*

Type	Multicast Group IP address	Group Advisor IP address
Multicast Group Sequence Number	Hop count	RTC indicator

For example, when a replacement timer expires, the group advisor (i.e., node 1 in Fig. 3), broadcasts GAN message, it sets the Request to Change (RTC) indicator. This indicator is an additional field in GAN message to indicate that the group advisor finishes its mission and wants to leave its place as a group advisor. The main fields of GAN message are shown in Table (2). A tree node (Green node) that receives GAN message checks RTC indicator. When RTC is set, tree/group node unicasts a *report message* to the group advisor carrying its energy status (so it is called energy report).

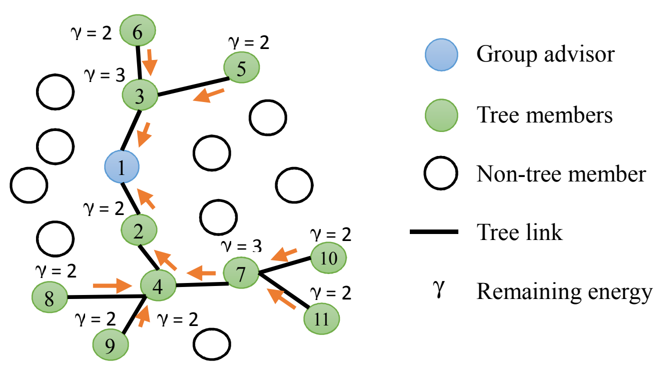


Fig. 3. *Group advisor receives energy report message from tree member nodes.*

Table (3) shows the main fields in an energy report message. When a tree/group node has not taken its place as a group advisor in the current time interval, it examines its current energy and updates the *member energy* field. As shown in Table (3), additional field is considered, energy report (ER) indicator field. Such field is set by tree/group node to inform the group advisor that it does not take a place as a group advisor in the current time interval and the member energy field carries its current node energy. Figure 3 shows that the group advisor receives energy report messages from the tree member nodes. After the group advisor receives the *report* messages from its member nodes, it selects the node (3) that has the highest residual energy to be the new group advisor. Then the group advisor unicasts NGA message to selected node as a *New Group Advisor (NGA)*. The NGA message is a type of report message that has additional field for assigning a new group advisor (it is called NGA indicator field).

Table 3. *Main fields of energy report message.*

Type	Destination IP Address	Destination Sequence Number
Hop count	Member energy	ER indicator

When the new group advisor receives NGA message, it changes Group Advisor IP address to its address and resets the hop count, then starts to broadcast a GAN message. In order to group members can detect such GAN message from new group advisor, the GRP_U indicator field in GAN message is set. Once the group member nodes receive that message, it updates its multicast information. The group advisor replacement procedure is repeated until all multicast tree members become group advisors.

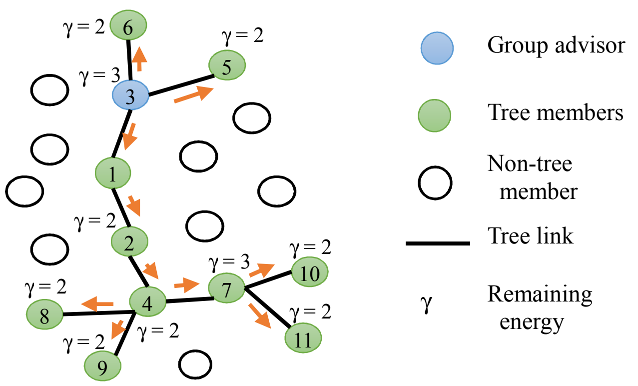


Fig. 4. *Multicast tree after changing the group advisor.*

The *replacement timer* is settled at reasonable value in which it is not very large resulting in a tree node depletes its energy, and is not very small resulting in high communication overhead. Figure 4 shows the multicast tree after the new group advisor is selected.

3.2.2. Tree Conservation

BAMUR scheme exploits the on-demand routing principles and conserve the multicast tree by two methods: clipping the tree whenever a node leaves the multicast group, and healing a damaged link. The healing method includes relinking the tree after the network was separated and rebuilding branches when a link fails.

i. Clipping Multicast Tree

For the duration of the regular operation of the network, a member of a multicast group might choose to leave the multicast group. If the node is not a leaf node of a multicast tree it might cancel its member status, but it must continue to work as a router for the tree. On the other hand, in case of the node is a leaf node it might clip itself from the tree. It creates a startup message and sets the clip indicator, then unicasts startup message to next hop (the next hop towards the group advisor). If the next hop node is not a member of the multicast group, and became a leaf node, it can also clip itself from the tree. The clipping process stops when either a multicast group member node is reached (regardless that node is a leaf node or not) or a non-leaf node is reached (regardless that node is a group member or not).

ii. Healing Damaged Links

The healing process can be described in three steps: detecting the damaged link, choosing a new group advisor for the network segment without an advisor, and merging the two separated parts of the network (sub-trees) as follows.

A multicast group tree links connectivity might be damaged due to node movement or the route expiration timer is terminated. The detection of a link damage is defined by absence of received packets from the neighbor at the time equals:

$$\text{announcement Interval} \times (1+ \text{acceptable Announment Loss})$$

where *announcement Interval* is the time between two announcement messages and the *acceptable Announcement Loss* is number of times of resending an announcement message. When a route is expired and a link damage is observed, the node that is further away from group advisor heals the damaged link.

The farthest node starts to select a new group advisor for this part of the network. After a new multicast advisor is chosen, the new group advisor sets the update indicator in a group announcement message and broadcasts it through its connected part of the network. The update indicator indicates a new group advisor existence; accordingly, all nodes should update their query table and their multicast route table.

When a node receives a GAN message from a multicast group having a group advisor differs from the multicast group it belongs to, it detects the existence of two separate parts. In such case, this node chooses one of the two group advisors to be the advisor for the two groups. The chosen group advisor sets the update indicator and sends a Group Announcement message. When all members in the other part (including the group advisor itself) receives such message, they know that there is a new group advisor and the tree is conserved by merging two sub-trees.

4. Performance Evaluation and Simulation Results

In this section, BAMUR performance is compared with MAODV and PDTMRP and the simulation results are conducted using Network Simulator 2 (NS2) [20]. The simulation area is 500 m × 500 m area with 50 mobile nodes. The simulation time is 800 seconds to be enough to study the changes in behavior of the network over a long time, especially the effect of energy consumption of the nodes on network characteristics. Also, for simulation purposes, the initial node energy is chosen equal 10J with the threshold energy 1J. We assume the communication range is 50 meters and the data packet size is 250 bytes. The number of source and receiver nodes is randomly chosen from 5 to 10 and from 15 to 25, respectively to effectively study the behavior of the multicast group. The traffic load is also randomly chosen from 5 to 25 packet/sec to study the process of selecting the routes between sources and multicast groups. List of simulation parameters are listed in Table (4).

Table 4. *Simulation parameters.*

Parameter	Value
Simulation area	500 m × 500 m
Simulation time	800 seconds
Number of nodes	50 nodes
Wireless communication range	50 m
Queue size	50 packets
Data packet size	256 bytes
Traffic load	5~25 packets/sec
Number of senders	5~10 nodes
Number of receivers	15~25 node

In what follows, the impact of traffic load, number of nodes and path loss for MAODV, PDTMRP and BAMUR schemes on WSN performance. The WSN performance can be studied using certain measures of effectiveness such as network lifetime, throughput, packet delivery ratio, end-to-end delay, and network control overhead. Such measures of effectiveness are defined as follows.

- Network lifetime is defined as the time until the first dead sensor node appears. Since the simulation time is about 800 seconds, the network lifetime parameter is measured in seconds.
- Network throughput is defined as the average amount of successful delivered packets (in bits) from all senders during the simulation time. Network throughput parameter is measured in kbps.
- Packet delivery ratio is the ratio between the total number of successful delivered packets and the total number of transmitted packets.
- End-to-end delay is the average time a data packet will take to reach the destination. Such parameter is measured by milliseconds.
- Network control overhead is the number of control packets sent divided by the number of data packets received.

4.1. The Impact of Traffic Load

Set of experiments are performed to compare the network performance for MAODV, PDTMRP and BAMUR schemes at different traffic loads. Fifty sensor nodes are randomly distributed in the target field and move according to random waypoint mobility model. Assume such nodes move in an urban environment with path loss 3.1 with zero noise.

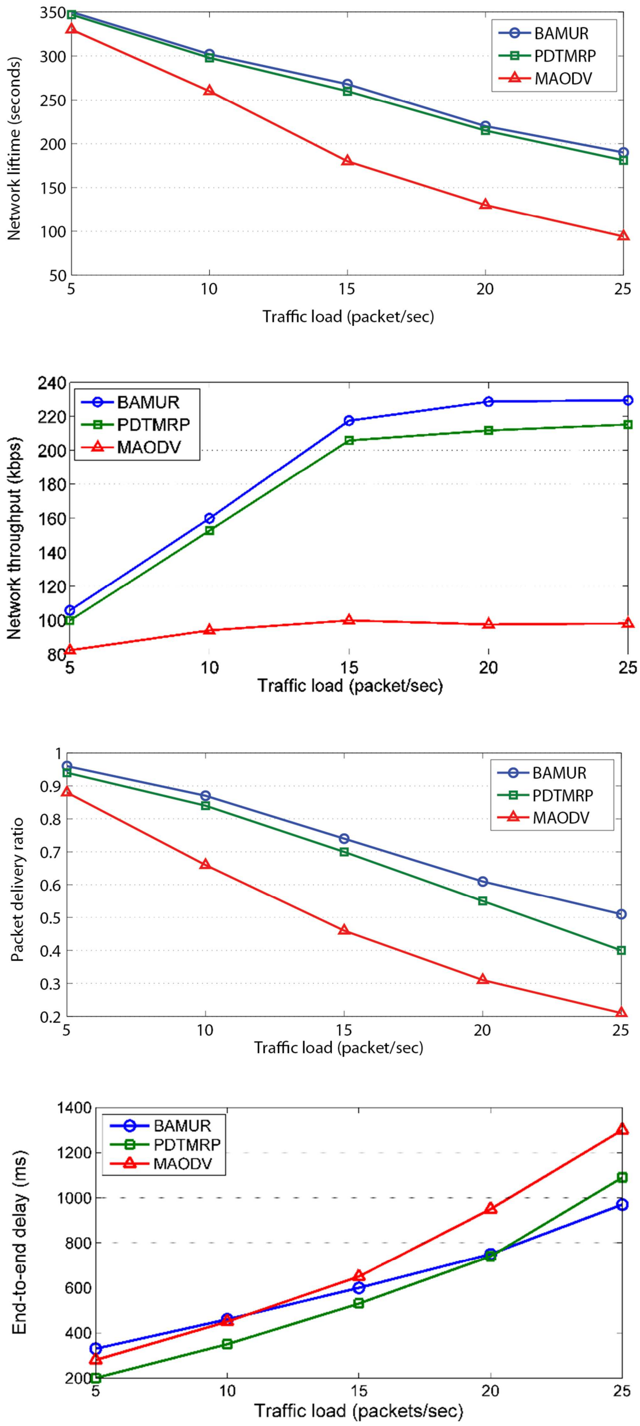


Fig. 5. *The impact of traffic load.*

(a) Network lifetime (b) Network throughput (c) Packet delivery ratio (d) End-to-end delay

At low traffic load (i.e., traffic load equals 5 packet/sec), as shown in Fig. 5 (a), the network lifetime in all schemes is approximately the same which reach 330, 347 and 350 seconds, respectively. When the traffic load increases to reach 10 packet/sec, the performance of PDTMRP and BAMUR schemes is better than MAODV scheme which the network lifetime reaches 298 and 302 seconds in PDTMRP and BAMUR schemes while it reaches 260 seconds in MAODV scheme. At traffic load equals 15 packet/sec, the performance of PDTP and BAMUR schemes are still better than MAODV scheme which the network lifetime reaches 260 and 268 seconds in PDTMRP and BAMUR schemes while it reaches 180 seconds in MAODV scheme. Briefly, PDTMRP and BAMUR schemes are better than MAODV scheme in all different traffic loads; however, BAMUR scheme slightly outperforms PDTMRP scheme.

For example, at traffic load equals 25 packet/sec, the lifetime difference reaches nine seconds between BAMUR and PDTMRP schemes. Consequently, the results show that BAMUR scheme outperforms other schemes in higher traffic loads. Figure 5 (b) shows the network throughput for MAODV, PDTMRP, and BAMUR at different traffic loads. At traffic load equals 5 packet/sec, the network throughput reaches 82.4, 100 kbps for MAODV, PDTMRP while it reaches 105.9 kbps for BAMUR scheme. When the traffic load increases to reach 10 packet/sec, the throughput increases, which reaches 94.1 and 152.9 kbps in MAODV and PDTMRP schemes while it reaches 161.5 kbps in BAMUR scheme. At traffic load equals 25 packet/sec, BAMUR and PDTMRP obviously outperform MAODV which throughput reaches 229.6 and 215.3 kbps in BAMUR and PDTMRP while it reaches 98.2 kbps in MAODV. As shown in Fig. 5 (b), the throughput in BAMUR scheme is better than it in MAODV and PDTMRP schemes because it achieves more energy saving due to balancing metrics in BAMUR stages (multicast foundation and replacing group advisors) resulting in a less number of dead nodes.

Figure 5 (c) shows the packet delivery ratio (PDR) for MAODV, PDTMRP, and BAMUR at different traffic loads. At traffic load equals 5 packet/sec, PDR reaches 0.88, 0.94 for MAODV, PDTMRP while it reaches 0.96 for BAMUR scheme. When the traffic load increases to reach 10 packet/sec, the PDR decreases, which reaches 0.66 and 0.84 in MAODV and PDTMRP schemes while it reaches 0.87 in BAMUR scheme. At traffic load equals 25 packet/sec, BAMUR scheme clearly outperforms other schemes which PDR reaches 0.51, while it reaches 0.4 and 0.21 in PDTMRP and MAODV. As shown in Fig. 5 (c), the packet delivery ratio in BAMUR scheme is better than it in MAODV and PDTMRP schemes because it achieves a minimum number of dropped packets due to prolonging the network lifetime.

Figure 5 (d) shows end-to-end delay for MAODV, PDTMRP, and BAMUR at different traffic loads. At low traffic load (5 packet/sec), PDTMRP achieves less end-to-end delay which reaches 205 msec because it depends on dual-tree structure that reduces route reconstruction. Therefore, it reduces the control overhead, achieving minimum end-to-end delay. Otherwise, end-to-end delay in MAODV and BAMUR schemes reaches 282 and 334 msec, respectively. The hop counts for chosen routes in BAMUR scheme may increase compared with MAODV at low traffic load to balance nodes energy. When the traffic load reaches 10 packets per seconds, a little bit nodes are missed in MAODV forcing it to choose routes relatively have the same hop counts in BAMUR schemes; therefore, end-to-end delay reaches 462 and 456 msec in BAMUR and MAODV schemes, respectively. On the other hand, PDTMRP scheme still achieves less end-to-end delay which reaches 356 msec. When the traffic load reaches 15 packets per seconds, more nodes are missed in MAODV while BAMUR scheme prolongs node lifetime due to its balanced metrics. Accordingly, end-to-end delay in BAMUR scheme is better than it in MAODV as shown in Fig. 4 (d).

When traffic load increases to 20 packet per seconds, end-to-end delay in BAMUR scheme approaches to its value in PDTMRP scheme. At high traffic load (i.e., 25 packets per seconds), BAMUR scheme achieves more balancing of node energy keeping a large number of nodes alive compared with PDMRP scheme. Shortest routes will be available for BAMUR schemes achieving less end-to-end delay as shown in Fig. 4 (d). Briefly, BAMUR scheme achieves less end-to-end delay compared to other schemes at higher traffic loads.

4.2. The Impact of Node Density

In this section, the network performance for MAODV, PDTMRP and BAMUR schemes is compared at different number of nodes.

The results are conducted at traffic load equals 10 packets per second, which the random waypoint mobility model is used and sensor nodes move in an urban environment with path loss 3.1 with zero noise. As shown in Fig. 6 (a), when fifty sensor nodes are used in the target field, the network lifetime parameter (i.e., the time of first dead sensor node event) reaches 298 and 302 seconds in PDTMRP and BAMUR schemes while it reaches 260 seconds in MAODV scheme.

Therefore, when sixty sensor nodes are used, network lifetime reaches 328 seconds in the BAMUR scheme while it reaches 310 and 277 seconds in PDTMRP and MAODV schemes, respectively. The number of sensor nodes gradually increases to reach 100 nodes and the results, as shown in Fig. 6 (a), indicates that BAMUR scheme outperforms other schemes in the network lifetime. For example, at one-hundred sensor nodes, the network lifetime in BAMUR scheme reaches 565 seconds while it reaches 506 and 325 seconds in PDTMRP and MAODV schemes, respectively.

Figure 6 (b) shows the impact of node density of the network throughput. At fifty nodes, the network throughput reaches 94.1 and 152.9 kbps in MAODV and PDTMRP schemes while it reaches 161.5 kbps in BAMUR scheme due the proposed balanced metrics that keep sensor nodes alive for longer times. When the number of nodes increases, all schemes approximately remain at the same level of network throughput.

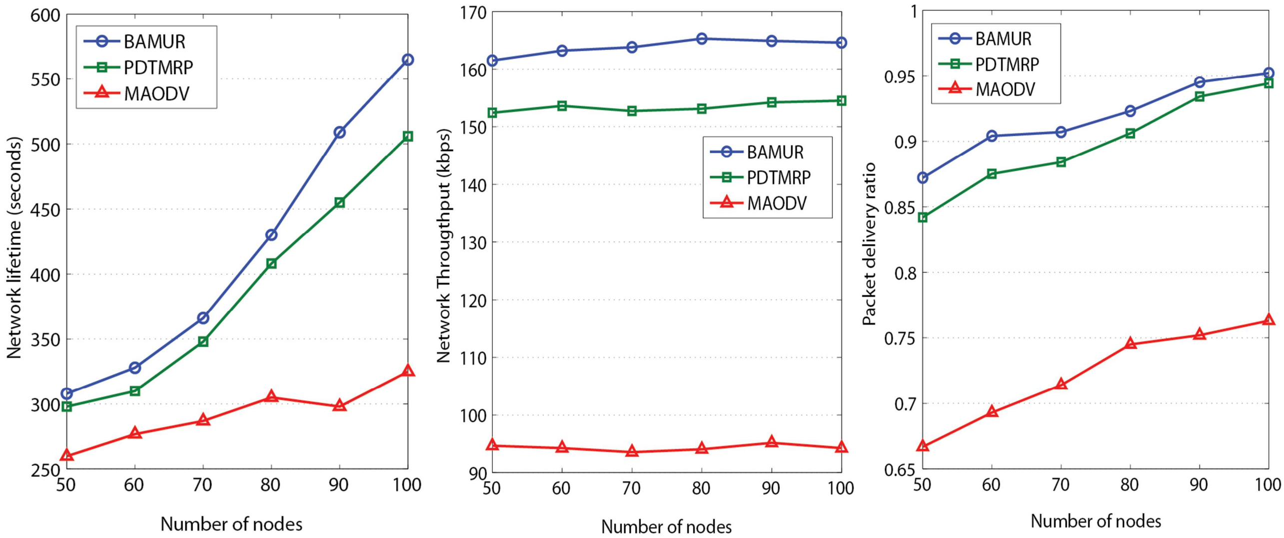


Fig. 6. *The impact of node density.*

(a) Network lifetime (b) Network throughput (c) Packet delivery ratio

Accordingly, BAMUR scheme achieves better network throughput regardless the number of deployed nodes. The impact of node density on the packet delivery ratio (PDR) is shown in Fig. 6 (c). When the number of sensor nodes increase, the packet delivery ratio increases in all schemes because a sender can easily discover multiple routes to receivers. BAMUR scheme achieves better packet delivery ratio at any number of nodes, as shown in Fig. 6 (c), because its main strategy is to prolong the network lifetime by the proposed balanced metrics.

4.3. The Impact of the Noise Level

In this section, the network performance for MAODV, PDTMRP and BAMUR schemes is compared at different values of noise level in two types of environments (urban and rural regions). The results are conducted at traffic load equals 10 packets per second which the random waypoint mobility model is used and the number of sensor nodes is 50. Assume the path loss in urban and rural regions are 3.1 and 5.1, respectively. In practice, signals in free space suffer from many effects such as multipath fading and shadowing. Accordingly, a noise component is added to path loss which varies from zero to 30% to study its impact on the packet delivery ratio.

Figure 7 (a) shows the impact of noise level on the packet delivery ratio (PDR) when a sensor network is deployed in urban region. When neglecting environmental effects by setting zero noise and path loss exponent equals 3.1, the PDR reaches 0.66 and 0.84 in MAODV and PDTMRP schemes while it reaches 0.87 in BAMUR scheme. When noise level increases, a sensor node cannot discover all neighbors in its transmission range. Certain neighbors will be congested because a sensor node is forced to choose those neighbors in its constructed route. Therefore, the probability of packet loss in entire network increases. As shown in Fig. 7 (a), when a noise level reaches 10%, the PDR reaches 0.61 and 0.78 in MAODV and PDTMRP schemes while it reaches 0.83 in BAMUR scheme. When a noise level reaches 20%, the PDR reaches 0.55 and 0.72 in MAODV and PDTMRP schemes while it reaches 0.77 in BAMUR scheme. At 30% of noise level, the PDR reaches 0.51 and 0.69 in MAODV and PDTMRP schemes while it reaches 0.75 in BAMUR scheme.

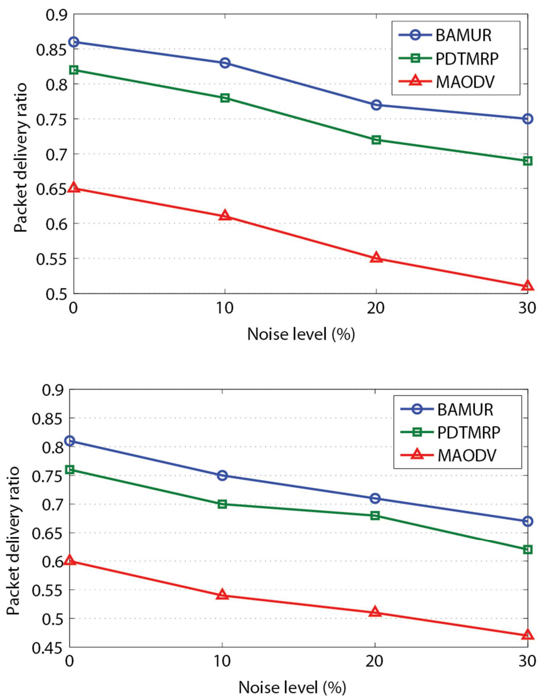


Fig. 7. *The impact of noise level at different environments.*

(a) Packet delivery ratio in urban regions (b) Packet delivery ratio in rural regions

Similarly, the impact of noise level on the packet delivery ratio (PDR) when a sensor network is deployed in rural region. When neglecting environmental effects by setting zero noise and path loss exponent equals 5.1, the PDR reaches 0.6 and 0.76 in MAODV and PDTMRP schemes while it reaches 0.81 in BAMUR scheme. Since rural regions are noisy more than urban regions, more packet loss is expected. As shown in Fig. 7 (b), when a noise level reaches 10%, the PDR reaches 0.54 and 0.7 in MAODV and PDTMRP schemes while it reaches 0.75 in BAMUR scheme. When a noise level reaches 20%, the PDR reaches 0.51 and 0.68 in MAODV and PDTMRP schemes while it reaches 0.71 in BAMUR scheme. At 30% of noise level, the PDR reaches 0.47 and 0.62 in MAODV and PDTMRP schemes while it reaches 0.67 in BAMUR scheme.

4.4. The Impact of Mobility Models

Since sensor mobility plays an important role in network control overhead, the impact of different mobility models on the network control overhead is studied in this section. Simulation results are conducted at traffic load equals 10 packets per second and the number of sensor nodes is 50 in urban regions with path exponent equals 3.1 with zero noise. Four mobility models are chosen to evaluate the network control overhead parameters: Gauss Markov (GM), Manhattan, Random Way Point (RWP) and Reference Point Group Mobility (RPGM) models. Two snapshots are taken from our simulation experiments at 400 and 800 seconds as shown in Fig. 8.

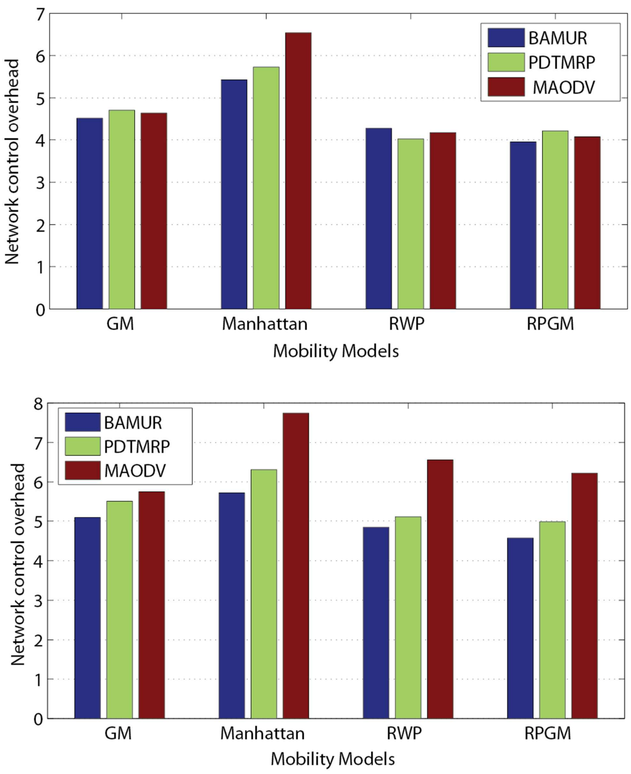


Fig. 8. *The impact of mobility models at 400 and 800 seconds of simulation time.*

(a) Network control overhead at 400 seconds (b) Network control overhead at 800 seconds

As shown in Fig. 8 (a), a snapshot is taken during the simulation time (around middle of simulation time). All sensor nodes are still alive in all schemes at this moment; therefore, the network control overhead approximately at the same level for all schemes. When Gauss Markov model is used, network control overhead in BAMUR, PDTMRP and MAODV reaches 4.52, 4.71 and 4.64, respectively.

While network control overhead in BAMUR, PDTMRP and MAODV reaches 5.43, 5.73 and 6.54 in Manhattan mobility model. Network control overhead in Manhattan is greater than Gauss Markov because there is a rapid change in the network topology when a Manhattan mobility model is used. When Random Way Point (RWP) model is used, network control overhead in BAMUR, PDTMRP and MAODV reaches 4.27, 4.02 and 4.17, respectively. On the other hand, network control overhead in BAMUR, PDTMRP and MAODV reaches 3.95, 4.21 and 4.07 when RPGM mobility model is used.

The second snapshot is taken at the end of simulation time (at 800 seconds), as shown in Fig. 8 (b). Since our scheme achieves more energy saving due to the proposed balanced metrics, BAMUR scheme obviously achieves minimum control overhead. When Gauss Markov model is used, network control overhead in PDTMRP and MAODV reaches 5.51 and 5.75 while it reaches 5.1 in BAMUR scheme. The network control overhead in PDTMRP and MAODV reaches 6.31, and 7.74 in Manhattan mobility model, while it reaches 5.72 in BAMUR scheme. When Random Way Point (RWP) model is used, network control overhead in PDTMRP and MAODV reaches 5.12 and 6.56 while it reaches 4.84 in BAMUR scheme. Network control overhead in PDTMRP and MAODV reaches 4.98 and 6.22 when RPGM mobility model is used, while it reaches 4.57 in BAMUR scheme.

Finally, we can conclude that the BAMUR scheme performance is better than MAODV and PDTMRP schemes in terms of network lifetime, throughput, packet delivery ratio, end-to-end delay and network control overhead. At higher traffic loads, the BAMUR performance is clearly better than the existing schemes.

5. Conclusion

In this paper, BAMUR scheme is proposed to achieve balanced metrics for multicast routing in WSNs to prolong the network lifetime. The proposed scheme calculates the RS for all candidate routes in terms of residual energy of intermediate nodes and hop count to group advisor, then it selects the route with the highest RS. Furthermore, BAMUR scheme changes the group advisor node periodically every predefined replacement time interval. The impact of traffic load, node density, noise level, mobility models, and network control overhead on the system performance is studied. The simulation results show that BAMUR is an energy-aware multicast scheme because it prolongs the network lifetime at higher traffic loads more than MAODV and PDTMRP schemes. Hence, BAMUR scheme achieves better network throughput and packet delivery ratio, and obviously reduces end-to-end delay. Simulation results show that BAMUR scheme remains better than the other schemes when the number of sensor nodes increase in the target field. Although the network throughput in BAMUR scheme decreases when a noise level of signal propagation in urban or rural regions increases, it remains better than the throughput of other schemes. Finally, simulation results show that the network control overhead that required for multicast operations is reduced compared to other schemes at different mobility models. In the future, BAMUR scheme can be extended to impressively minimize the energy consumption and control overhead for route detection process.

References

[1] A. Pullin, and C. Pattinson, "A Realistic Battlefield Model for the Evaluation of MANET," *Wireless on Demand Network Schemes and Services,* pp. 81-84, 2008.

[2] R. Vaishampayan, and J. Garcia-Luna-Aceves, "Efficient and Robust Multicast Routing in Mobile Ad Hoc Networks," *IEEE International Conference on Mobile Ad-Hoc and Sensor Schemes,* p. 304–313, 2004.

[3] S. Lee, W. Su, and M. Gerla, "On-demand multicast routing protocol in multihop wireless mobile networks," *Mobile Networks and Applications,* vol. 7, no. 6, pp. 441-453, 2002.

[4] E. Royer, and C. Perkins, draft-ietf-manet-aodv-00-" Multicast Ad hoc On-Demand Distance Vector (MAODV) Routing," http://tools.ietf.org/html/draft-ietf-manet-maodv-00, 2000.

[5] N. Wang, "Power-aware dual-tree-based multicast routing protocol for mobile ad hoc networks," *IET Communications,* vol. 6, no. 7, p. 724–732, 2012.

[6] S. Bae, S. Lee, W. Su, and M. Gerla, "The design, implementation, and performance evaluation of the on-demand multicast routing protocol in multi-hop wireless networks," *IEEE Network: The Magazine of Global Internetworking,* vol. 14, no. 1, pp. 70-77, 2000.

[7] K. Oe, A. Koyama, and L. Barolli, "A Multicast Routing Protocol for Wireless Mesh Networks Considering Network Load," *Advanced Information Networking and Applications, IEEE 27th International,* pp. 591-597, 2013.

[8] J. Garcia-Luna-Aceves, and E. Madruga, "The core-assisted mesh protocol," *IEEE Journal on Selected Areas in Communications,* vol. 17, no. 8, pp. 1380-1394, 2006.

[9] J. Xie, R. Talpade, A. McAuley, and M. Liu, "AMRoute: Ad Hoc Multicast Routing Protocol," *Mobile Networks and Applications,* vol. 7, p. 429–439, 2002.

[10] V. Devarapalli, and D. Sidhu, "MZR: a multicast protocol for mobile ad hoc networks," *Communications, 2001. ICC 2001. IEEE International,* pp. 886-898, 2001.

[11] T. Pusateri, "Distance Vector Multicast Routing Protocol," in *Juniper Networks*, 2004.

[12] S. Moh, S. Lee, and C. Yu, "Adaptive multicast on mobile ad hoc networks using tree-based meshes with variable density of redundant paths," *Wireless Networks,* vol. 15, no. 8, pp. 1029-1041, 2009.

[13] C. Wu, and Y. Tay, "AMRIS: A multicast protocol for ad hoc wireless networks," in *IEEE Military Communications Conference MILCOM*, Atlantic City, USA, 1999.

[14] J. Jetcheva, and D. Johnson, "Adaptive Demand-Driven Multicast Routing in Multi-Hop Wireless Ad Hoc Networks," *ACM international symposium on Mobile ad hoc networking & computing,* pp. 33-44, 2001.

[15] C. Perkins, E. Royer, and S. Das, draft-ietf-manet-aodv-05.txt "Ad hoc On-Demand Distance Vector (AODV) Routing," https://tools.ietf.org/html/draft-ietf-manet-aodv-05, 2000.

[16] X. Wei., G. Chen, W. Yingyu, and F. Mtenzi, "Optimized priority based energy efficient routing algorithm for mobile ad hoc networks," *Ad Hoc Networks Journal,* vol. 2, no. 3, pp. 231-239, 2004.

[17] M. Morteza, and P. Massoud, "Lifetime-aware multicast routing in wireless ad hoc networks," *IEEE Wireless Communication and Networking,* vol. 3, pp. 1317-1323, 2004.

[18] NC. Wang, and Y. Su, "A power-aware routing protocol for mobile ad hoc networks with mobility prediction," *IEEE conference on Local Computer Networks,* pp. 408-418, 2005.

[19] R. Manoharan, P. Thambidurai, and S. Lakshmana-Pandian, "Energy efficient Robust On-Demand Multicast Routing Protocol for MANETs," *International Journal of Ad Hoc and Ubiquitous Computing,* vol. 3, no. 2, pp. 90-98, 2008.

[20] M. S. Gharajeh, "Avoidance of the energy hole in wireless sensor networks using a layered-based routing tree," *International Journal of Systems, Control and Communications,* vol. 7, no. 2, pp. 116-131, 2016.

[21] "The Network Simulator 2," [Online]. Available: http://www.isi.edu/nsnam/ns/.

Biography

Abdelrahman S. Halawa (abdelrahmanhalawa@gmail.com) currently works as a teaching assistant at Al-Azhar University and a member of NARSS (National Authority for Remote Sensing & Space Sciences) projects. He participated IEEE computer society international contest in 2009/2010 which achieved with his team the top prize (first place between more than 82 Universities all over the world) over the summer break of 2010. He graduated in 2011 with a Bachelor's of engineering degree in Systems and Computer Engineering. After graduation, he worked in CWS (Center for Wireless Studies) at Cairo University for three years. He was working in developing Tool for Extensive Management and Performance Optimization (TEMPO) for 2G and 3G mobile networks. His research interests include Cellular Networks, Wireless Sensor Networks, and Self Organized Networks.

Salah M. Abdel-Mageid (smageid@azhar.edu.eg) received his M.S. and Ph.D. in Systems and Computers Engineering from Al-Azhar University in 2002 and 2005, respectively. Since 2012, he is an associate professor in Systems and Computers Engineering at Al-Azhar University. He performed his post-doctoral research in 2007 and 2008 in the Computer Science and Engineering Department, School of Engineering, the Southern Methodist University at Dallas, TX, USA. He was a member of TEMPO (Tool for Extensive Management and Performance Optimization) project in Cairo University and Vodafone Egypt as an industrial partner in 2014 and 2015. His research interests include Mobile Computing, Cellular Networks, Sensor Networks, Cognitive Radio Networks, Vehicular Ad-hoc Networks, Big Data and Data Analysis, Internet Services and Applications.