IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received July 27, 2017, accepted August 16, 2017, date of publication August 30, 2017, date of current version October 12, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2746676

MSGR: A Mode-Switched Grid-Based Sustainable Routing Protocol for Wireless Sensor Networks

SURAJ SHARMA[1], DEEPAK PUTHAL[2], SABAH TAZEEN[1], MUKESH PRASAD[2], AND ALBERT Y. ZOMAYA[3], (Fellow, IEEE)

[1]Department of Computer Science and Engineering, International Institute of Information Technology at Bhubaneswar, Bhubaneswar 751003, India
[2]Faculty of Engineering and IT, University of Technology Sydney, Sydney, NSW 2007, Australia
[3]School of Information Technologies, The University of Sydney, Sydney, NSW 2006, Australia

Corresponding author: Deepak Puthal (deepak.puthal@uts.edu.au)

ABSTRACT A Wireless Sensor Network (WSN) consists of enormous amount of sensor nodes. These sensor nodes sense the changes in physical parameters from the sensing range and forward the information to the sink nodes or the base station. Since sensor nodes are driven with limited power batteries, prolonging the network lifetime is difficult and very expensive, especially for hostile locations. Therefore, routing protocols for WSN must strategically distribute the dissipation of energy, so as to increase the overall lifetime of the system. Current research trends from areas, such as from Internet of Things and fog computing use sensors as the source of data. Therefore, energy-efficient data routing in WSN is still a challenging task for real-time applications. Hierarchical grid-based routing is an energy-efficient method for routing of data packets. This method divides the sensing area into grids and is advantageous in wireless sensor networks to enhance network lifetime. The network is partitioned into virtual equal-sized grids. The proposed mode-switched grid-based routing protocol for WSN selects one node per grid as the grid head. The routing path to the sink is established using grid heads. Grid heads are switched between active and sleep modes alternately. Therefore, not all grid heads take part in the routing process at the same time. This saves energy in grid heads and improves the network lifetime. The proposed method builds a routing path using each active grid head which leads to the sink. For handling the mobile sink movement, the routing path changes only for some grid head nodes which are nearer to the grid, in which the mobile sink is currently positioned. Data packets generated at any source node are routed directly through the data disseminating grid head nodes on the routing path to the sink.

INDEX TERMS Wireless sensor networks, grid-based routing, grid head, mobile sink, energy efficiency.

I. INTRODUCTION

A Wireless Sensor Network (WSN) is a distributed network with small embedded devices having sensing capability called sensor nodes, which are used in huge numbers to observe physical or environmental conditions such as temperature, pressure, heat, humidity etc. from the environment [1]. The sensor nodes collect the variations in physical parameters and coordinate among themselves to relay the data through the sensor nodes to a base station or sink. A sensor node consists of sensing unit, a sensor to measure the physical stimuli, analog to digital converter, processing unit with a processor and storage area, a transceiver which can transmit and receive the data, and to run all devices a small battery is used. The sensor nodes are low power and low cost devices which make it appropriate to deploy them in a network in large scale. Deployment of sensor nodes in large numbers increases the coverage of the network and enhances the reliability of data transmission and retrieval. Utilization of sensor networks may be for environmental monitoring, smart homes and offices, surveillance, military applications and many more.

Sensor nodes have some constraints like inadequate battery and processing capability, low bandwidth collision-prone channels etc. Sensor nodes are often deployed in the hostile and unattended environment. These conditions do not allow replacement of the battery of the sensor nodes. It is necessary to improve the life of the sensor nodes. Thus, the protocols designed for this network must be energy efficient and distributed. There must be proper balance of the load between the sensor nodes, which point to the better lifetime the sensor network. There exist many hierarchical-based routing protocols, typical like cluster-based, and atypical like grid-based, chain-based, area-based and tree-based routing techniques [2].

Researchers have proposed several grid-based routing techniques. A virtual grid-like structure is created by the source node to route the data through selected data forwarding nodes on the grid towards one or more destination nodes. Geographic forwarding is used as sensor nodes know their location using GPS coordinates. In cases where the actual location is not known, nodes may use the virtual coordinates. Either a single node per cell in a grid, known as the coordinator node or four nodes per cell, called the data disseminating nodes, are used for data routing depending on their distance from the source or the amount of residual energy present in them. Thus, these nodes can effectively do load balancing in the network increasing the longevity of the sensor network. Once their energy depletes or if they fail for some reason, new nodes will be elected to serve the purpose. The destination node or the sink may be stationary or mobile in grid-based structures where mobile sinks are handled differently [3].

The contributions of this paper are summarized as follows.

1) Firstly, the whole sensing area is divided into virtual grids and followed by a grid head selected from individual grids.
2) Then, we follow the MSGR protocol for data packet transmission towards the mobile sink.
3) Finally it proposes a method to manage sink mobility and grid head re-election.

The rest of the paper is arranged as follows. Section II explains the related work. Section III outlines the proposed protocol. Section IV examines and analyzes the simulation results. Section V concludes the paper with future directions.

II. RELATED WORK

In 2005, Luo *et al.* [4] proposed TTDD which is based on grid architecture built by the source node whenever it senses an external stimuli and generates some data to send. This protocol considers that sensor nodes' locations are fixed and have their location information by using GPS or other means. These mobile sinks send queries in order to collect data from the source node. The source node starts building the grid structure without waiting for the sink to query. The data is forwarded recursively through special nodes on the grid called data dissemination nodes until the data reaches the sink. The grid is composed of square cells each of size $\alpha \times \alpha$. The grid construction starts with initially assuming that the source node is at one crossing point of grid say (x, y). The next four adjacent points are calculated as $x_i = \mathrm{x} + \mathrm{i}\alpha$ and as $y_j = \mathrm{y} + \mathrm{j}\alpha$ where α is the size of a cell of a grid and i, j $= \pm 0, \pm 1, \pm 2$.. The sensor nodes nearest to the dissemination points calculated previously are selected as dissemination nodes. The source node will broadcast a data announcement packet to all four adjacent grids. The receiver node stores the information sent by the source node and the announcement message is further forwarded to this node's next four adjacent grid points excluding the node from which it receives the message. This process repeats until the entire grid is built. Thus, propagation of data announcement messages selects dissemination nodes nearest to the dissemination points and each cell has four data dissemination nodes. The query from the sink is flooded initially inside the local cell till it reaches the nearest data dissemination node. This is lower-tier traversing of the query. Then, the query is forwarded to upstream dissemination nodes from which data announcement messages were received, who forward it towards the source. This is higher-tier traversing of the query. Thus, the query traverses two tiers to reach the source. The query is aggregated in case of multiple sinks requesting the same data. Similarly, the requested data sent by the source also follows the two-tier forwarding approach, but in reverse order to reach the sink in TTDD.

In 2006, a multicast routing protocol GMR was proposed by Sanchez *et al.* [5]. It is fully localized and works solely on the basis of information provided by neighbours. It delivers multicast data messages to one set of destination nodes efficiently without flooding the network. Each source node with data messages selects the best possible subset of its neighbours in terms of cost to move towards the destination. The cost is measured as the number of neighbours selected and progress is taken as the diminution in the distances left to destination nodes. GMR uses geographic routing where sensor nodes know their current location using GPS or other means and they inform their positions to neighbour nodes using periodic beacon signals. Thus, a source node gets the locations of its destination nodes beforehand. GMR models the network as a unit disk graph (UDG). GMR selects neighbours using greedy set partition selection algorithm where the number of destinations can be large.

In 2007, Buttyan and Schaffer [6] proposed PANEL in which the sensor nodes are present within a fixed area which is geographically partitioned into a number of clusters. PANEL elects the aggregator node within each cluster in the sensor network to which other sensor nodes within the cluster forward their sensor readings so that they are processed, combined and compressed at the aggregator node. Queries from the sink are sent to the aggregator of a cluster. Sensor nodes are time-synchronized where time is divided into various epochs and a different node gets elected as an aggregator node in each epoch in order to balance the network load. In PANEL, two different types of routing are done. One is intra-cluster routing, which is within the cluster to route messages already inside the cluster to the present aggregator node or to any of the previous aggregators; if the message is from a distant source, the other is inter-cluster routing which is between clusters. PANEL faces the problem of node depletion which may lead to election of more than one aggregator within a cluster when the connected-sub-network of a cluster gets partitioned.

In 2007, Akl and Sawant [7] proposed a Grid-based Coordinated Routing protocol, where any one node in each grid is elected to act as the coordinator node. The source node floods the network with its data and has a querying message for all the coordinator nodes, which take part in routing. As soon as

TABLE 1. Comparison of grid-based routing protocols.

Protocols	Control Manner	Mobility of Sink	Network Type	Routing Path Initiation	Time Synchronization	Application Scenario
TTDD	distributed	yes	proactive	event-based	not synchronized	multiple mobile sinks
GMR	distributed	yes	reactive	event-based	not synchronized	multicast application
PANEL	distributed	yes	proactive	query-based	synchronized	synchronous or asynchronous application
Grid-Based Coordinated Routing	distributed	no	proactive	query-based	not synchronized	single stationary sink
HRPM	distributed	yes	proactive	query-based	not synchronized	multicast application
HGMR	distributed	yes	proactive	query-based	not synchronized	multicast application
GMCAR	distributed	yes	proactive	query-based	not synchronized	QoS-based network
EAGER	distributed	yes	proactive	event-based	synchronized	multiple mobile sinks
VGDRA	distributed	yes	proactive	proactive	not synchronized	single mobile sink
VGBST	distributed	yes	proactive	proactive	not synchronized	single mobile sink
GBRR	distributed	yes	proactive	proactive	not synchronized	single base station

the sink node gets the data, flooding stops. Any information or query from the sink is transmitted to the source node using the reverse back route. This procedure repeats till any coordinator exhausts its energy. The source has to re-flood the network so that the sink can figure out some new route back to the source. This process stops when the connectivity between the source and sink no longer exists due to partitioning of the network. Nodes other than the coordinator nodes sleep by powering down their radio signals to conserve energy. Thus, overall energy gets conserved. A coordinator goes through three states before running out of energy. If its energy is greater than 25% it is still in routing state. When the energy level gets less than or equal to 25%, it is in warning state. It gets depleted when energy equals zero. The node having the largest ID is elected to be the coordinator in each grid. To ensure connectivity among coordinators in adjacent grids, the size upper bound on a square grid of width equal to r is r$\leq R_n/\sqrt{5}$ where R_n is the maximum transmitting distance.

In 2008, Das *et al.* [8] proposed a robust and scalable multicast routing protocol Hierarchical Rendezvous Point Multicast (HRPM) protocol. It incorporates two key design ideas. First, it hierarchically decomposes a huge group into a hierarchy of smaller subgroups. Second, HRPM uses the concept of distributed geographic hashing to build and maintain this hierarchy at no additional cost. The group members of the multicast tree conform to a fixed Rendezvous Point (RP) node as the group manager. HRPM efficiently manages the group membership and location of nodes. HRPM divides the sensor field into equal sized square cells until each cell has a manageable number of members and every cell has an Access Point (AP) to manage its members. HRPM limits the per-packet encoding overhead while routing data packets to some constant and incurs minimum tree encoding overhead while partitioning the group into subgroups. The source builds a virtual Src →AP tree. The packets of data are sent to the Src→AP tree. The AP then routes the data to the remaining Src→AP tree. The AP builds an AP→member overlay tree and send packets to the group members. Holes in HRPM are handled using face routing. For holes encountered during routing to a hashed location, a sequenced number of packets is utilized.

In 2008, Koutsonikolas *et al.* [9] proposed a new location aware routing, named protocol Hierarchical Geographic Multicast Routing Protocol (HGMR) for static sensor networks. It takes into consideration the design principles of GMR and HRPM providing both forwarding efficiency and reduced encoding overhead giving an energy-efficient and scalable multicast protocol. In HGMR, for reducing encoding overhead, a hierarchy of subgroups is constructed similar to HRPM. For data delivery, for source→AP tree, HRPM's unicast method is used which provides reliability. For AP→member tree, GMR's broadcast-based forwarding is used where the number of multicast group members is large which significantly reduces the number of transmissions. In HGMR, the source→member overlay tree is similar to that used in HRPM. The AP→member trees in each cell comprises some destination nodes. Using GMR's localized neighbour selection method these destination nodes are selected. These trees are not overlay trees as in the case of HRPM.

In 2012, Banimelhem and Khasawneh [10] proposed a grid-based multipath routing protocol named GMCAR. The GMCAR protocol also includes additional features by avoiding network congestion to support QoS traffic routing in WSN. Cross layer architecture is always suitable for network congestion control [11]. Initially, the sensor network is partitioned into square-shaped grids. Every grid has a number of nodes and one single master node. GMCAR protocol maintains many diagonal routes through every master node of each grid and the base station. The base station creates and sends a flooding message which reaches each grid so that the master nodes will find routes from their grid towards the sink. Since non-boundary grids have high traffic, there is more than one diagonal paths available to route the incoming packets towards the sink. This lets the data packets travel the minimum number of hops in reaching the sink. Boundary grids having lower traffic have one horizontal or vertical path towards the sink. The master node routes the data received to the next suitable master node. If the master node runs out of energy, a new master node is elected based on the residual energy of nodes. When the number of data packets at the buffer of a master node crosses a threshold, a congestion avoidance and congestion mitigation mechanism is initiated. A secondary master node is elected which shares the traffic in a congested grid in order to mitigate the congestion. GMCAR uses two separate routing schemes for low traffic and high traffic which conserves energy leading to higher network lifetime. GMCAR also considers QoS which gives higher throughput.

In 2013, Chi and Chang [12] proposed an energy-aware grid-based routing technique named EAGER for WSN. A virtual grid is constructed and each grid has a unique Grid Identification (GID.) A node calculates the grid to which it belongs using GID(X, Y) $= \{(\mathrm{x}, \mathrm{y}) | \mathrm{x} = \lfloor (X - x_0)/\alpha \rfloor$, $\mathrm{y} = \lfloor (Y - y_0)/\alpha \rfloor\}$. A node in each grid is elected as the Grid Head which maintains the list of adjacent Grid Heads. This protocol applies a time-scheduling technique and keeps Grid Heads whose sum of coordinates are also active. If it is odd, the radio is turned off for a defined time interval determined by the scheduling technique. Time unit is divided into 2^n timeslots and a constant timeslot number is assigned for sleep schedule. Time slot number $= [(\mathrm{GID.X} + \mathrm{GID.Y}) \bmod 2^n + \mathrm{GID.X} \bmod 2^{n-1}]$. It ensures that all Grid Heads are always in active state with any set of four adjacent grids. Source's Local Grid Head floods a REQ packet to build the routing path. Sink's LGH replies with the REP packet which reaches the source's LGH. Thus, data are transmitted along this path. When the sink proceeds to a different grid, it extends the path to reach the sink and uses rerouting to build a shorter path to reach the sink.

In 2015, Khan *et al.* [13] proposed VGDRA. VGDRA partitions the network into virtual grids consisting of uniformly-sized cells. The set of nodes closest to the centre of a cell are selected as cell-headers. The gateway nodes are elected for the communication between the adjacent cell-headers. Cell-headers construct a virtual backbone structure together with the gateway nodes to keep information about the current position of the sink. The member nodes associate with the nearest cell-header for data communication. The cell-header collects the data from the member nodes and relays them towards the mobile sink using the path. The mobile sink moves around the network to collect the data from the border cell-header. The border cell-header closest to the sink is responsible for route re-adjustment.

In 2016, Sharma and Suresh [14] proposed VGBST, where the virtual backbone structure comprises of a set of cell headers designated for reconstructing the new routes based on the current position of the sink. The sensor field is partitioned into a virtual grid of uniform sized cells for designing virtual infrastructure. Cell-headers are appointed based on the sensor nodes near to the center of the cells that keeps track of the mobile sink's latest location. Apart from cell headers, others nodes transmit their data to the nearest cell-header. The cell-header forwards the data to its adjacent cell-headers through gateway nodes.

In 2016, Meng *et al.* [15] proposed Grid-Based Reliable Routing (GBRR). GBRR creates virtual on square grids in which the next communication hop is chosen based on communication quality. GBRR partitions a two dimensional WSN into equal square-shaped grids, so that there could be zero or some sensor nodes in one grid. Using the current location of nodes and grids as the basis of the clustering algorithm, overall energy consumption is saved rather than calculating the whole complicated network topology. One cluster may occupy one grid or more, and a cluster head is elected to be the active node which takes the ability of controlling intra-cluster and inter-cluster communication. In order to avoid overloading of head nodes, the routing algorithm calculates the most effective paths along and in the clusters, so that the source does not need to transmit information to the BS through the path with head nodes on the way. One cluster may occupy a grid or some, and a cluster head is elected to be the active node which takes the ability of controlling intra-cluster and inter-cluster communication. In order to avoid overloading of head nodes, the routing algorithm calculates the most effective paths along and in the clusters, so that the source does not need to transmit information to the BS through the path with head nodes on the way. The summarize of all the related protocols discussed is listed in Table 1.

While considering real-time applications using IoT, fog or edge computing by collecting data streams from sensors, security and privacy of the data play a vital role [16]. There are lots of solution available to protect sensor data streams or data packets from several cyber threats [17]–[20]. In our previous work, we have divided security solutions into CIA triode i.e. confidentiality, integrity and availability by considering security threats [17], [18].

III. PROPOSED MSGR PROTOCOL

The proposed routing protocol, Mode-Switched Grid-based Routing (MSGR) reduces the flow of control packets and incorporates techniques to enhance the network lifetime. In the previously proposed protocol EAGER, REQ packet (Request Control packet) is flooded to find the path to the sink and REP (Reply Control packet) is sent by the sink through the shortest of paths from which REQ packets were received. In managing the random movement of mobile, a considerable amount of control overhead occurs. EAGER checks for any new possible shorter path than the current path which also causes control overhead. In MSGR, the sink initiates the routing path formation. Once a sink gets its location, the routing path is maintained using fewer exchanges of control packets. The overhead of calculating time-slot is avoided in MSGR. The modes of Grid Heads involved in the routing process are changed alternately in order to have balanced consumption of node energy. So, after a definite time period, the nodes which were idle earlier will now initiate routing while the previous set of nodes will go inactive. MSGR aims to reduce the overhead of rerouting. Random sink movement has less impact as only a few nodes get altered along the routing path in MSGR.

The sensor nodes and sink are aware about their geographical locations. The sensor nodes in the network are stationary and their clocks are synchronized. The sensor nodes are capable of turning their radio channel on or off when required in a synchronized fashion. A single mobile sink is able to collect data from different sources at any random time. Sensor nodes are homogeneous in nature with an initial uniform energy level.

A. CONSTRUCTION OF GRID

To construct a virtual grid infrastructure of sensor nodes in the entire sensor field, MSGR uses a pair of numbers to identify the Grid Identification known as GID, which identifies each grid as shown in Figure 1. The sensors belonging to the same grid compute the same GID using their GPS location co-ordinates. Before deploying the sensor nodes, grid size α is set to a predefined value (say, 20). Using the transmission range R_{tr} the grid size α is calculated, where $\alpha = R_{tr}/(2\sqrt{2})$. Thus, a node in one grid can communicate directly with sensor nodes in its eight adjacent grids through the radio channels.

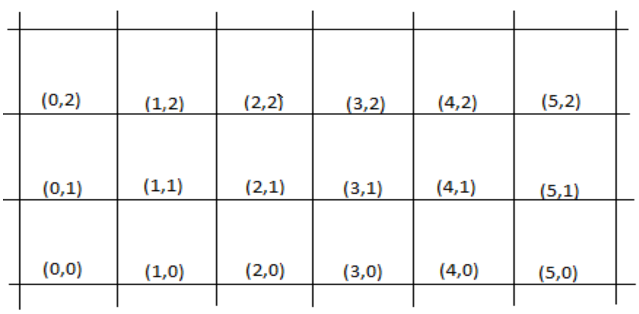

FIGURE 1. Grid construction.

Each node calculates its Grid ID (GID) using the geographic location coordinates(X,Y):

$$\text{GID(X, Y)} = \{(grid_x, grid_y) | grid_x = \lfloor (X)/\alpha \rfloor, \quad grid_y = \lfloor (Y)/\alpha \rfloor\}$$

B. GRID HEAD ELECTION

In MSGR, certain data disseminating nodes are elected to route data from the source node to the mobile sink. One node in each grid is elected for routing data, called the Grid Head, in a random manner by the following procedure. Initially, all the sensor nodes have uniform battery power. Each node in every grid invokes its associated timer randomly. Within a grid, the node which timeouts the earliest gets selected as the Grid Head of that grid and notifies other members of the grid of its election. The member nodes on receiving the notification from this selected node cancel their timers and select this node as their Grid Head. Since other members do not take part in routing, they keep their sensing channel on and turn off their radio until they sense any stimuli generated from an external event.

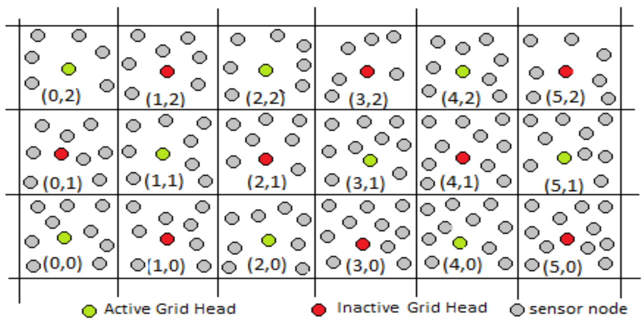

FIGURE 2. Grid head election.

Algorithm 1 Mode Setting of Grid Heads

$grid_x$: x co-ordinate of the grid of the Grid Head
$grid_y$: y co-ordinate of the grid of the Grid Head
GH_MODE:A Grid Head node operation either active(1) or sleep mode(0)
t: timer associated with each Grid Head for mode change
for(each Grid Head)
  if($(grid_x + grid_y) \bmod 2 == 0$)
   GH_MODE ← 1
  else
   GH_MODE ← 0
  endif
endfor

In MSGR, initially, those Grid Heads whose sum of $grid_x$ and $grid_y$ co-ordinates are even are made active, i.e., their radio channel is on for a defined time interval (t) and their GH_MODE is set as 1 (active mode). Those Grid Heads whose sum of $grid_x$ and $grid_y$ co-ordinates gives an odd number sleep for the same time period t and set their GH_MODE as 0 (sleep mode), refer algorithm 1 and Figure 2. At the end of time interval t, the Grid Head modes are swapped. This switching of modes between active and sleep takes place throughout the lifetime of the sensor network. This helps to save Grid Heads energy when they are idle and also helps in uniform distribution of network load.

C. SINK DETECTION

Sink broadcasts a SINK_LOCATION packet which contains its $grid_x$ and $grid_y$ co-ordinates. The Grid Head on receiving this packet sends a BEACON message with its $grid_x$ and $grid_y$ coordinates to the sink. Sink on receiving BEACON packet, checks for $grid_x$ and $grid_y$ values of the packet. If it matches with the Sink's $grid_x$ and $grid_y$ coordinates, then sink sends an ACK message to the Grid Head. The Grid head then sets its next_hop towards the sink and becomes the Sink's Local Grid Head (LGH). Sink drops any more BEACON messages from adjacent Grid Heads. Then, this Local Grid Head broadcasts SINK_DETECTION packet. This packet contains the Origin Grid Head node's Grid ID. The Origin Grid Head node is the node which broadcasts this packet. The adjacent Grid Heads which are active receive this packet and set their next_hop towards the source of this packet. The SINK_DETECTION packet is then rebroadcast to be received by the four adjacent Grid Heads which also set their next_hop in the same way. At the end of this phase, all the active Grid Heads shall have formed the routing path which reaches towards the sink. This phase is described in algorithm 2 and shown in Figure 3 and Figure 4.

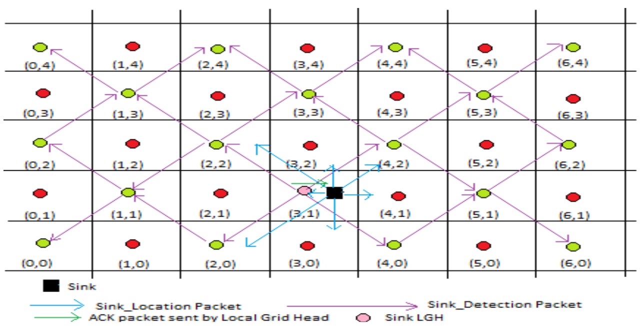

FIGURE 3. Broadcasting of Sink_Location by the sink and Sink_Detection packets by active grid heads.

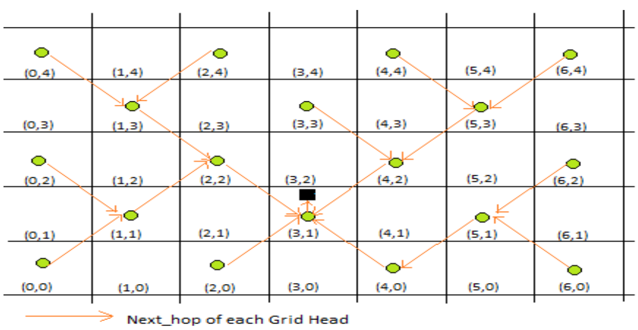

FIGURE 4. Next hop of a grid head set in accordance with the sink location.

D. DATA TRANSMISSION

When any sensor node detects any target event, it collects the data and becomes the source to send data. First, it

---

Algorithm 2 Sink Detection

SINK_LGH: Sink's Local Grid Head
$sink.grid_x$: $grid_x$ of sink
$sink.grid_y$: $grid_y$ of sink
Grid Heads receive SINK_LOCATION packet from sink
SINK_LOCATION: $< sink.grid_x, sink.grid_y >$
sink receives BEACON packet from Grid Heads
BEACON:$< grid_x, grid_y >$
GH_id: selected Local Grid Head ID
A Grid Head receives ACK packet from sink
ACK:$< GH_{id} >$
$Origin\_GH.grid_x$: $grid_x$ of Source GH node which broadcasts this packet
$Origin\_GH.grid_y$: $grid_y$of source GH node which broadcasts this packet
Adjacent Grid Heads receive SINK_DETECTION packet from SINK_LGH
SINK_DETECTION:$< Origin\_GH.grid_x, Origin\_GH.grid_y >$
sink_selected: initialized as false. Set to true if any GH selects next_hop as sink
$GH.grid_x$ & $GH.grid_y$: x & y co-ordinates of the grid of a Grid Head
next_hop:next Grid Head node towards which the routing path is set. Initially set to NULL.
flag: to ensure next_hop is not changed by another Grid Head in the same iteration. Initially set to false.
$sink.grid_x \leftarrow floor(X/\alpha)$
$sink.grid_y \leftarrow floor(Y/\alpha)$
sink broadcasts SINK_LOCATION packet
for(each receiver Grid Head)
  send a BEACON packet to the sink
if(ACK received from the sink)
  $SINK\_LGH \leftarrow$ self
  $next\_hop \leftarrow$ sink
  $sink\_selected \leftarrow$ true
endif
end for
Sink_LGH broadcasts SINK_DETECTION packet
for(each adjacent receiver Grid Head)
  x1 $\leftarrow Origin\_GH.grid_x$
  y1 $\leftarrow Origin\_GH.grid_y$
if(($GH.grid_x$ == x1 − 1&&$GH.grid_y$ == y1 − 1) ||($GH.grid_x$ == x1 − 1&&$GH.grid_y$ == y1 + 1)||($GH.grid_x$ == x1 + 1&&$GH.grid_y$ == y1 + 1) ||($GH.grid_x$ == x1 + 1&&$GH.grid_y$ == y1 − 1))
  if(sink_selected == false && flag == false && next_hop == NULL)
    next_hop $\leftarrow Origin\_GH$
    flag $\leftarrow$ true
    Rebroadcast SINK_DETECTION packet
    Repeatfor
  endif
endif
end for

broadcasts a META_DATA packet. Upon receiving the META_ DATA packet, the receiver Grid Head nodes send a META_DATA_ACK packet. If the META_DATA_ACK packet reaches the source node from the Local Grid Head node (LGH), then, data is sent to the LGH which forwards the data to its next hop Grid Head, which again forwards it in the same manner until the data reach the sink. If the Local Grid Head of source was in sleep mode, then adjacent active Grid Heads receive the META_DATA packet. When the first META_DATA_ACK packet from any one of the four adjacent Grid Head nodes reaches the source node, the data is transmitted to that adjacent Grid Head node which further relays it to its next hop Grid Head until the sink receives the data from its Local Grid Head. The source node drops any more META_DATA_ACK packets received later. At any point of time, any four adjacent Grid Heads will always be active according to the way Grid Head modes are set. The data transmission is described in algorithm 3 and shown in Figure 5.

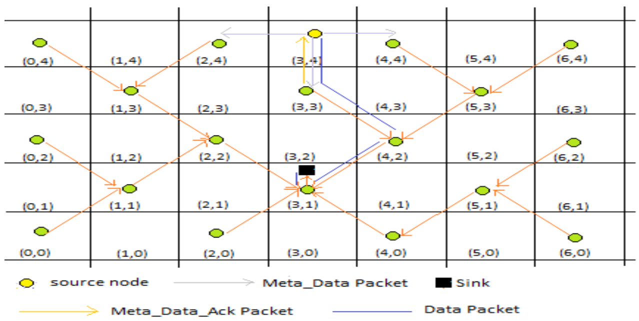

FIGURE 5. Data transmission from source to sink.

Lemma 1: Optimum multipath routes reduce data forwarding delay.

Proof: MRP uses optimal path. Let the distance from the target node be d, length of the data packet is L, bandwidth$B$, light speed is c and the processing and queuing time is T_{extra}, Then in the primary path the data forwarding delay time can be represented by

$$T_{primary} = \sum \frac{d}{c} + \sum d\frac{L}{B} + \sum T_{extra} \tag{1}$$

Thus, with respect to distance d the delay will be decreased.

Lemma 2: The entire message complexity of individual network is in the order $O(nk)$.

Proof: Let, n be the number of sensors deployed in the sensing area. $O(k)$is the complexity of the neighbouring phase of a sensor node, where k implies quantity of neighbours. Considering multipath configuration, let 'p' be the primary nodes and 'a' be alternate nodes, where $(p + a) < n$. $O(3p + 2a)$ represents message complexity of the primary and alternate nodes. Primary node practices one broadcast message including two unicast messages, whereas alternate node utilizes one broadcast message including one unicast message. The routing protocol is utilizing 'p' number of messages for the route reply. $(nk + 3p + 2a + p)$ represent total messages in the network. Therefore, the entire message complexity of a particular network is $O(nk)$.

Algorithm 3 Data Transmission

LGH:Grid Head of grid in which source node lies
ADJ_GH:Grid Head with co-ordinates (x + 1, y + 1), (x − 1, y − 1), (x + 1, y − 1) or (x − 1, y + 1) if source node is in(x,y)
$source_{id}$:source node ID
Grid Heads receive META_DATA packet from source node
META_DATA:<META_DATA, $source_{id}$>
GH_{id}:ID of Grid Head node
source node receives META_DATA_ACK from Grid Head
META_DATA_ACK:<META_DATA_ACK,GH_{id}>
if(id == $source_{id}$)
  Broadcast META_DATA packet
  if(LGH sends a META_DATA_ACK packet)
   Send data to LGH
  else
   send data to ADJ_GH which first sent META_DATA_ACK packet
  endif
endif
for(each next_hop GH_{id})
  if (next_hop == Sink)
   send data to Sink
  endif
  break
  if(next_hop! = Sink && next_hop! = NULL)
   send data to next_hop GH_{id}
  endif
endfor

Lemma 3: If m is the number of sensors present in the optimal path then $O(m)$is the complexity of sending a packet from source to destination.

*Proof:*The MRP builds optimal multipath routing between source and destination. An unique path is used at each iteration for data transmission. Every node will reroute the data. In different words, individual sensors collect the data from the previous node, process and convey them to the subsequent sensors. Therefore, the time complexity is $O(m)$.

E. HANDLING SINK MOBILITY

The mobile sink is aware of the location and regularly broadcasts SINK_LOCATION packet. The current Grid Head may either be in an active mode or in sleep mode. So, when the sink receives BEACON packet from Adjacent Grid Heads, it suggests that its current Local Grid Head is in sleep mode. So, sink sends ACK packet to one of the Adjacent Grid Heads and makes it its Local Grid Head. So, the Local Grid Head of sink may lie in the adjacent grid or in the grid itself depending on the sink location and the current mode of the Grid Heads. The Local Grid

Algorithm 4 Handling Sink Mobility

Origin_GH:Grid Head node from which SINK_DETECTION packet is received
sink broadcasts SINK_LOCATION packet
for(each receiver Grid Head)
  send a BEACON packet to the sink
end for
if(ACK received from the sink)
  SINK_LGH $\leftarrow$ self
  next_hop $\leftarrow$ sink
  sink_selected $\leftarrow$ true
endif
Sink_LGH broadcasts SINK_DETECTION packet
for(each adjacent receiver Grid Head)
x1 $\leftarrow$ *Origin_GH*.$grid_x$
y1 $\leftarrow$ *Origin_GH*.$grid_y$
  if(($GH.grid_x$ == x1 − 1&&$GH.grid_y$ == y1 − 1) ||($GH.gridx$ == x1 − 1&&$GH.grid_y$ == y1 + 1)||($GH.grid_x$ == x1 + 1&&$GH.grid_y$ == y1 + 1) ||($GH.grid_x$ == x1 + 1&&$GH.grid_y$ == y1 − 1))
   if(next_hop! = *Origin_GH*)
    next_hop $\leftarrow$ *Origin_GH*
    Rebroadcast SINK_DETECTION packet
   endif
  endif
endfor

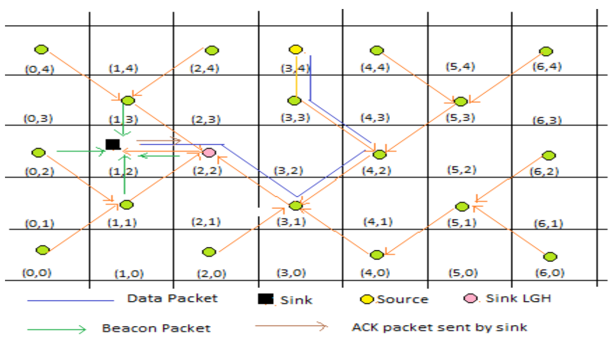

FIGURE 6. Handling sink mobility. Next hop changes w.r.t mobile sink.

Head sets its next_hop location towards the Sink. Then, the new Local Grid Head broadcasts SINK_DETECTION packet. The adjacent active Grid Heads upon receiving the packet checks whether the source Grid Head node of the SINK_DETECTION packet is different from their current next_hop. If it is same, the SINK_DETECTION packet is not forwarded further and this phase ends. If the source Grid Head of SINK_DETECTION packet is different at the receiver Grid Head, then the next_hop of the receiver Grid Head is updated towards the new source and SINK_DETECTION packet is rebroadcast as described in algorithm 4 and shown in Figure 6. The adjacent receiver Grid Heads perform the same operation and the packet is again rebroadcast until the source of this packet and next_hop of any Grid head is found to be similar.

Lemma 4: The entire energy consumption of the network can be represented by $E_{total} = \sum (E_{TX}(k, d) + E_{RX}(k) + E_{sleep}(t))$

Proof: Transmitting, receiving, and sleeping are the primary operations in MRP. Let consider $E_{TX(k,d)}$ be the energy for transmitting k bit message over distance d. For the same bit over distance d the reception energy is $E_{RX}(k)$ and it consumes $E_{sleep}(t)$ energy for sleep mode for t seconds. The following equation represents the entire energy consumed in the network.

$$E_{Total} = \sum \left(E_{TX}(k, d) + E_{RX}(k) + E_{sleep}(t)\right) \tag{2}$$

Algorithm 5 Switching of Grid Head Mode

for each Grid Head
if($grid_x + grid_y$ mod 2 == 0)
  if(GH_MODE == 1)
   if(next_hop! = 0)
    set state to sleep mode
    next_hop $\leftarrow$ 1
    GH_mode $\leftarrow$ 0
   else
    ▷ Even LGH remains in active mode
   endif
  else
   set state to active mode
   GH_MODE $\leftarrow$ 1
  endif
else ▷ for odd grid head
  if(GH_MODE == 0)
   set state to active mode
   GH_mode $\leftarrow$ 1
    ▷ Odd Grid head node switches to active mode
   else
    if(next_hop! = 0)
    set state to sleep mode
    next_hop $\leftarrow$ 1
    GH_MODE $\leftarrow$ 0
    ▷ Odd Grid head node goes to sleep mode
    else
     ▷Odd Grid Head remains in active mode
    endif
  endif
endif
endfor

F. MODE SWITCHING OF GRID HEAD

After every predefined time interval t, the modes of Grid Heads are swapped between active and sleep mode as described in algorithm 5 and shown in Figure 7. The switching of mode causes uniform dissipation of Grid Head node energy. Thus, it helps in balancing of network load and increases the throughput of the network. After a time interval

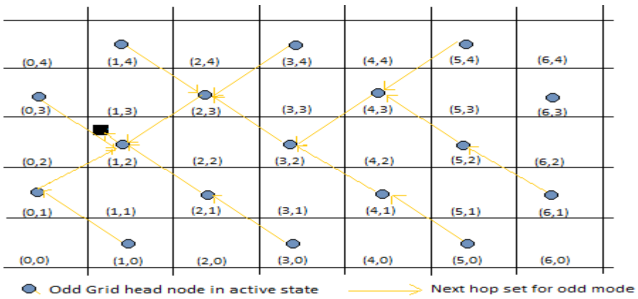

FIGURE 7. Mode switching of grid head. Next hop of odd grid set according to sink location.

of t, the next_hop field values of active Grid Heads are checked to determine if they contain the mobile sink. If the radio channel is free and the next_hop does not point to the sink, then the active Grid Heads set their next_hop to NULL and switch their modes to sleep mode and sleep for time t. That Grid Head whose next_hop points towards the sink remains active. Then, those Grid Heads which were initially in sleep mode switch to active mode. After the mode switch, the routing path establishment phase starts with the sink's LGH broadcasting SINK_DETECTION packet to adjacent active Grid Heads. Continuing with the procedure, next_hop values are set at each active Grid Head and the routing path is established followed by the data transmission from any possible source. Data transmission in odd mode are shown in Figure 8.

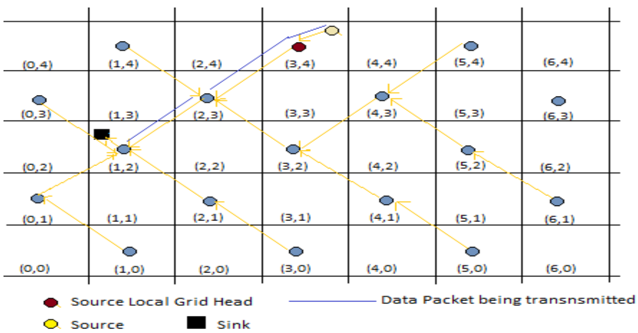

FIGURE 8. Data transmission involving odd grid heads after mode switch.

Lemma 5: The network lifetime is $\min\left\{\frac{TE}{Ec_i}\right\}$, where $i = 1, 2, 3, \ldots n$.

Proof: The total number of packets a node can communicate before dying is called the network lifetime. When battery power is over the node dies. Let the ith sensor node S_i be given the energy TE from this it uses E_{ND} for neighbour discovery,E_{MP} for building multipath, E_{DATA} for transmission and E_{PROC} for rest of the activities.

Then for a uniform network the network lifetime in MRP is represented by

$$\min\left\{\frac{TE}{Ec_i}\right\}$$

where,

$$Ec_i = E_{ND_i} + E_{MP_i} + E_{DATA_i} + E_{PROC_i} \tag{3}$$

Lemma 6: From node i to node j, $E_{ij(reliable)} = \frac{E_{ij}}{1-p_{ij}}$ represents the expected energy for reliable transmission of a packet.

Proof: In MRP, let E_{ij} be the energy to transmit a packet from node i to node j with packet error probability p_{ij} Hence $(1 - p_{ij})$ is error-free packet transmission and $\frac{1}{1-p_{ij}}$ number of packet retransmissions required from node i to node j Hence from node i to j for a reliable transmission expected energy requred is

$$E_{ij(reliable)} = \frac{E_{ij}}{1 - p_{ij}} \tag{4}$$

G. GRID HEAD RE-ELECTION

If the current energy of any Grid Head node falls below the threshold value, a timer gets triggered upon which all the member nodes become awake. After that, the Grid Head broadcasts a re-election notification packet. One of the member nodes which first receives this packet selects itself as the new Grid Head and notifies other members of the grid. The other member nodes select the new grid head and go back to sleep mode. The new Grid Head sends an UPDATE_HOP packet to its four diagonally adjacent Grid head nodes containing the node ID of old Grid Head. The Grid Heads on receiving this packet update their next_hop Grid Head to be the newly elected Grid Head if their next_hop field was set as the old Grid Head.

IV. EXPERIMENT AND EVALUATION

The proposed protocol MSGR is compared with the existing protocol EAGER over the four network parameters, such as network lifetime, packet delivery ratio, end-to-end latency and average energy consumption. The obtained results are plotted on the graph and their analysis is done. For simulation, the Castalia-3.2 simulator tool is used. It is based on the OMNeT++ platform. The simulation parameters listed in Table 2 are used for simulating the existing and the proposed protocol.

TABLE 2. Simulation parameters.

S.No.	Simulation Parameters	values
1.	Simulator used	Castalia Simulator (version 3.2)
2.	Network area	100 x 100 metres
3.	Number of nodes	200 nodes
4.	Sink speed	(5,10,15,20,25)secs
5.	Mobility model	Gaussian Mobility model
6.	Simulation time	120 secs
7.	MAC protocol	Tunable Mac
8.	Initial energy of nodes	10 Joules

A. NETWORK LIFETIME

Network lifetime is the duration of time when the first node dies in the network. The simulation result of network lifetime is shown in Figure 9. As the figure shows, the proposed protocol MSGR has more nodes alive as compared to the existing protocol EAGER in the given time span. This is becaus, MSGR saves more energy in nodes by switching data

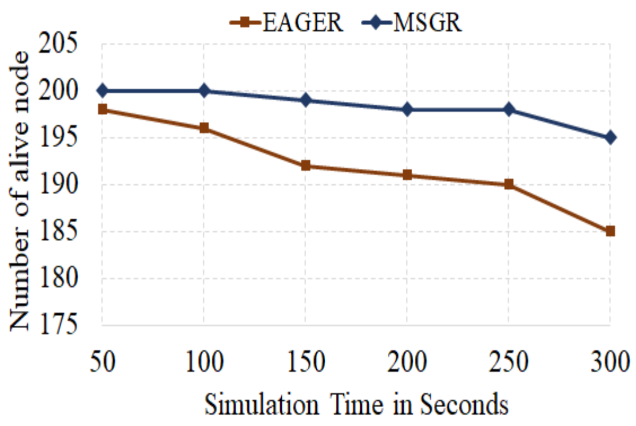

FIGURE 9. Comparison of network lifetime.

disseminating nodes to sleep state or active state alternately. The rerouting overhead in EAGER also causes more loss of node energy.

B. PACKET DELIVERY RATIO

The ratio is the percentage of data packets received successfully by the sink. The performance of packet delivery ratio decreases with the increase in speed of the sink. Packet delivery ratio of Proposed MSGR and existing EAGER is shown in Figure 10. As the sink moves faster, the grid and the local grid head of sink changes frequently. Also the number of hops from the source to sink increases, thus decreasing the packet delivery ratio. MSGR performs slightly better than EAGER.

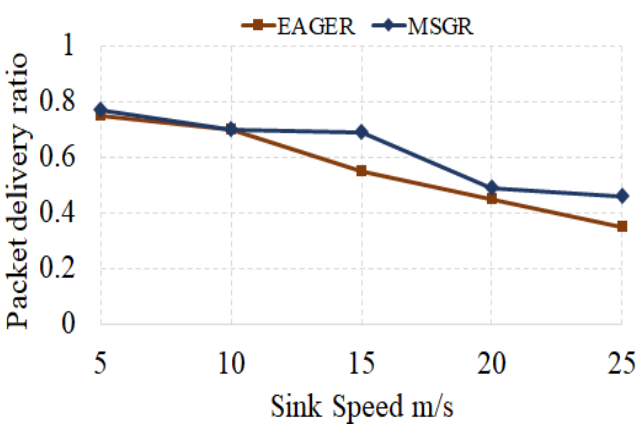

FIGURE 10. Packet delivery ratio.

C. END-TO-END DELAY

The end-to-end delay is the time between the source generating the data packet and that packet being successfully received at sink. The result of average end-to-end delay decreases with the increase in sink speed shown in Figure 11. The sink speed varies between 5 m/s and 25 m/s. As the sink changes its grid either in even mode or odd mode, it has possibilities of finding shorter routes through adjacent grid head nodes when the sink's LGH GID sum is the opposite

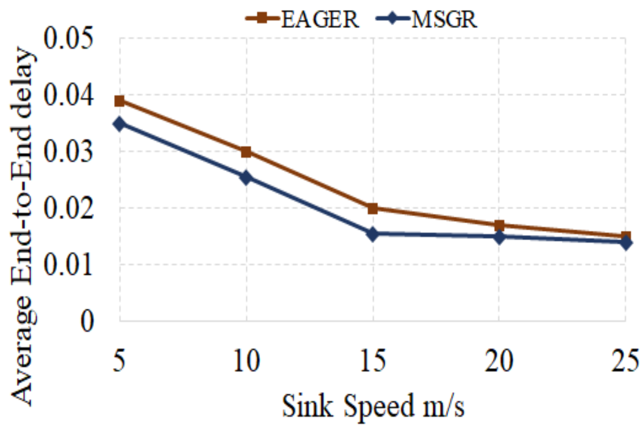

FIGURE 11. Average end-to-end delay.

of the current mode (even/odd). Since the gaussian mobility model is used, the end-to-end delay decreases with random sink movement. EAGER has higher end-to-end delay as the routing path becomes longer due to increasing sink speed.

D. AVERAGE ENERGY CONSUMPTION PER NODE

This is the average energy consumed by each node in the network due to reception and transmission of control and data packets. Average energy consumed per node is much lower in the proposed MSGR due to alternate switching of grid-head states as shown in Figure 12, and avoidance of flooding of control packets for building routing path using a reactive approach unlike EAGER, and avoidance of rerouting periodically to find the shorter path. In accordance with the approach followed, EAGER consumes more energy per node as compared to the proposed method.

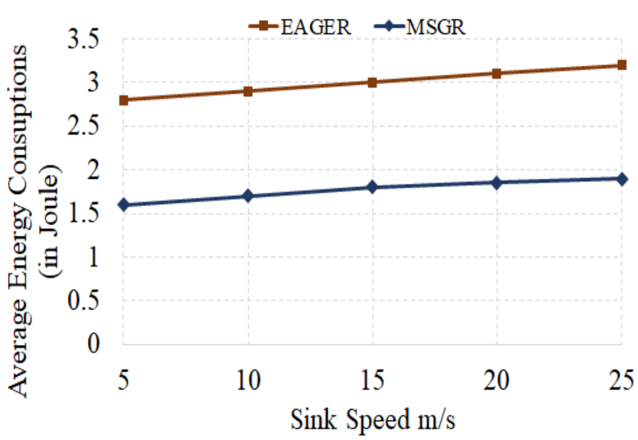

FIGURE 12. Average energy consumption.

V. CONCLUSION AND FUTURE DIRECTIONS

In this paper, a novel Mode-Switched Grid-based Routing protocol has been unveiled. It is capable of increasing the lifetime of the network through energy efficiency, and also improves the delivery delay for a single mobile sink. The simulation results of MSGR have been compared with

EAGER and the results confirm that MSGR performs better than EAGER over various network parameters and more effectively handles routing of data packet towards mobile sink. In the proposed MSGR protocol, the routing path is proactively built by setting the next hop of each Grid Head leading towards the sink. EAGER builds the routing path in a reactive manner whenever a source node is ready with data through flooding of REQ packets which consumes more energy. As the sink moves to a different grid, only some Grid Heads change their next hop Grid Head in MSGR. This results in lower consumption of energy in an already energy-constrained sensor network. In EAGER, the sink has to find the nearest next hop to build the extended path to the source when sink moves to a different grid. MSGR is free from rerouting overhead as the optimal path is already set whereas EAGER does rerouting to find the optimal path. In MSGR, the idle grid heads are allowed to sleep for specific intervals. The idle grid heads may be from odd sum GIDs or even sum GIDs due to rotation of grid head modes. This approach of mode switching Grid Heads in an alternate manner for a fixed interval in MSGR helps to balance consumption of network energy and increases the lifetime of WSN.

On examining the recommended protocol with the existent protocol, the proposed protocol MSGR gives better results in terms of four parameters, network lifetime, end-to-end delay, packet delivery ratio, average energy consumption. This is attributed to the fact that the proposed protocol uses a proactive approach in building the routing path. Once the network infrastructure is in place, the routing path is initiated involving data disseminating nodes. On the other hand, the existing protocol builds the routing path on demand through flooding of control packets. This causes more consumption of energy.

In future we are planning to extend this work by using multiple mobile sinks. It may lead to more flow of control packets, in which case this issue needs to be addressed.

ACKNOWLEDGMENT

This paper was presented at the International Conference on Computing, Communication and Automation 5–6 May, 2017. [21]

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, 2002.
[2] X. Liu, "Atypical hierarchical routing protocols for wireless sensor networks: A review," *IEEE Sensors J.*, vol. 15, no. 10, pp. 5372–5383, Oct. 2015.
[3] S. Sharma, D. Puthal, S. K. Jena, A. Y. Zomaya, and R. Ranjan, "Rendezvous based routing protocol for wireless sensor networks with mobile sink," *J. Supercomput.*, vol. 73, no. 3, pp. 1168–1188, 2017.
[4] H. Luo, F. Ye, J. Cheng, S. Lu, and L. Zhang, "TTDD: Two-tier data dissemination in large-scale wireless sensor networks," *Wireless Netw.*, vol. 11, nos. 1–2, pp. 161–175, 2005.
[5] J. A. Sanchez, P. M. Ruiz, and I. Stojmnenovic, "GMR: Geographic multicast routing for wireless sensor networks," in *Proc. 3rd Annu. IEEE Commun. Soc. Conf. Sensor Ad Hoc Commun. Netw. (SECON)*, Sep. 2006, pp. 20–29.
[6] L. Buttyán and P. Schaffer, "PANEL: Position-based aggregator node election in wireless sensor networks," in *Proc. 4th IEEE Int. Conf. Mobile Adhoc Sensor Syst. (MASS)*, Oct. 2007, pp. 1–9.
[7] R. Akl and U. Sawant, "Grid-based coordinated routing in wireless sensor networks," in *Proc. 4th IEEE Consum. Commun. Netw. Conf.*, Jan. 2007, pp. 860–864.
[8] S. M. Das, H. Pucha, and Y. C. Hu, "Distributed hashing for scalable multicast in wireless ad hoc networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 3, pp. 347–362, Mar. 2008.
[9] D. Koutsonikolas, S. M. Das, Y. C. Hu, and I. Stojmenovic, "Hierarchical geographic multicast routing for wireless sensor networks," *Wireless Netw.*, vol. 16, no. 2, pp. 449–466, 2010.
[10] O. Banimelhem and S. Khasawneh, "GMCAR: Grid-based multipath with congestion avoidance routing protocol in wireless sensor networks," *Ad Hoc Netw.*, vol. 10, no. 7, pp. 1346–1361, 2012.
[11] D. Puthal, Z. H. Mir, F. Filali, and H. Menouar, "Cross-layer architecture for congestion control in vehicular ad-hoc networks," in *Proc. Int. Conf. Connected Veh. Expo*, 2013, pp. 887–892.
[12] Y.-P. Chi and H.-P. Chang, "An energy-aware grid-based routing scheme for wireless sensor networks," *Telecommun. Syst.*, vol. 54, no. 4, pp. 405–415, Dec. 2013.
[13] A. W. Khan, A. H. Abdullah, M. A. Razzaque, and J. I. Bangash, "VGDRA: A virtual grid-based dynamic routes adjustment scheme for mobile sink-based wireless sensor networks," *IEEE Sensors J.*, vol. 15, no. 1, pp. 526–534, Jan. 2015.
[14] S. Sharma and D. Suresh, "VGBST: A virtual grid-based backbone structure type scheme for mobile sink based wireless sensor networks," in *Proc. Int. Conf. Adv. Res. Comput. Sci. Eng. Technol. (ICARCSET)*, 2015, p. 21.
[15] X. Meng, X. Shi, Z. Wang, S. Wu, and C. Li, "A grid-based reliable routing protocol for wireless sensor networks with randomly distributed clusters," *Ad Hoc Netw.*, vol. 51, pp. 47–61, Nov. 2016.
[16] D. Puthal, S. Nepal, R. Ranjan, and J. Chen, "Threats to networking cloud and edge datacenters in the Internet of Things," *IEEE Cloud Comput.*, vol. 3, no. 3, pp. 64–71, May/Jun. 2016.
[17] D. Puthal, S. Nepal, R. Ranjan, and J. Chen, " DLSeF: A dynamic key-length-based efficient real-time security verification model for big data stream," *ACM Trans. Embedded Comput. Syst.*, vol. 16, no. 2, p. 51, 2017.
[18] D. Puthal, X. Wu, S. Nepal, R. Ranjan, and J. Chen, "SEEN: A selective encryption method to ensure confidentiality for big sensing data streams," *IEEE Trans. Big Data*, to be published.
[19] D. Puthal, S. Nepal, R. Ranjan, and J. Chen, "A secure big data stream analytics framework for disaster management on the cloud," in *Proc. 18th IEEE Int. Conf. High Perform. Comput. Commun.*, Dec. 2016, pp. 1218–1225.
[20] A. Nanda, P. Nanda, X. He, A. Jamdagni, and D. Puthal, "Secure-GLOR: An adaptive secure routing protocol for dynami wireless mesh network," in *Proc. 16th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Aug. 2017, pp. 269–276.
[21] S. Sharma and S. Tazeen, "Mode-switched grid-based routing for wireless sensor networks," in *Proc. Int. Conf. Comput., Commun. Autom. (ICCCA)*, May 2017, pp. 1–6.

SURAJ SHARMA received the Ph.D. degree from the National Institute of Technology at Rourkela, Rourkela, India. He is currently an Assistant Professor with the Department of Computer Science and Engineering, International Institute of Information Technology at Bhubaneswar, Bhubaneswar. His research interest includes Internet of Things and wireless sensor networks.

DEEPAK PUTHAL received the Ph.D. degree in computer and information systems from UTS, Australia. He is currently a Lecturer (Assistant Professor) with the School of Computing and Communications, University of Technology Sydney, Australia. He has authored in several international conferences and journals, including IEEE and ACM transactions. His research interests include cyber security, Internet of Things, distributed computing, and big data analytics. He is an Associate Editor of the *IEEE Consumer Electronics Magazine* and the *KSII Transactions on Internet and Information Systems*. He also served as a Co-Guest Editor of several reputed journals, including the *Concurrency and Computation: Practice and Experience*, the *Wireless Communications and Mobile Computing*, and the *IEEE Consumer Electronics Magazine*.

SABAH TAZEEN received the B.E. degree from Savitribai Phule Pune University in 2013. She is currently pursuing the M. Tech. degree with the International Institute of Information Technology at Bhubaneswar, Bhubaneswar. Her research interests include wireless communications, networking, and in routing in wireless sensor networks.

MUKESH PRASAD received the master's degree in computer application from Jawaharlal Nehru University at Delhi, New Delhi, India, in 2009. He is currently pursuing the Ph.D. degree with the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan. He is currently a Lecturer (Assistant Professor) with the School of Software, University of Technology Sydney, Australia. He has authored several journal and international conference papers. His current research interests include machine learning, big data, pattern recognition, fuzzy systems, and neural networks.

ALBERT Y. ZOMAYA (F'04) is currently the Chair Professor of high performance computing & networking and an Australian Research Council Professorial Fellow with the School of Information Technologies, The University of Sydney. He is also the Director of the Center for Distributed and High Performance Computing which was established in 2009. His research interests include the areas of parallel and distributed computing and complex systems. He was a recipient of the IEEE Technical Committee on Parallel Processing Outstanding Service Award in 2011, the IEEE Technical Committee on Scalable Computing Medal for Excellence in Scalable Computing in 2011, and the IEEE Computer Society Technical Achievement Award in 2014. He is a Chartered Engineer, and a fellow of the AAAS and IET. He served as the Editor-in-Chief of the IEEE TRANSACTIONS ON COMPUTERS from 2011 to 2014. He was elected recently as a Founding Editor-in-Chief for the newly established the IEEE TRANSACTIONS ON SUSTAINABLE COMPUTING. He also serves as an Associate Editor of 22 leading journals, such as the *ACM Computing Surveys*, the *ACM Transactions on Internet Technology*, the IEEE TRANSACTIONS ON CLOUD COMPUTING, and the IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS. He is the Founding Editor of several book series, such as the Wiley Book Series on Parallel and Distributed Computing and Springer Scalable Computing and Communications.