International Journal of Computer (IJC)

ISSN 2307-4523 (Print & Online)

http://ijcjournal.org/

GADMS: Gathering Aggregated Data using Mobile Sink in Wireless Sensor Networks

E. G. Prathima[a]*, Laxmikant[b], Naveen S. A.[c], Venugopal K. R.[d], S. S. Iyengar[e], L. M. Patnaik[f]

[a,b,c,d]University Visvesvaraya College of Engineering, Bangalore University, Bangalore, India

[e]Florida International University, USA

[f]INSA, National Institute of Advanced Studies, Indian Institute of Science Campus, Bangalore, India

[a]Email: eprathima@yahoo.co.in

Abstract

WSNs consist of resource constrained sensor nodes that monitor the physical environment and transmit their data to the Sink through multi-hop communication. Mobile sinks are used to reduce the number of hops the data travels and thereby reducing the overall energy consumption. In this paper we propose Gathering Aggregated Data using Mobile Sink in Wireless Sensor Networks (GADMS) protocol that allows the mobile sink to collect data from WSNs where path of the mobile sink is not predetermined. The mobile sink halts at a point in the network and broadcasts an aggregate query. The average path length of a data packet is a constant and hence it can withstand node failures. The performance analysis shows that GADMS incurs less energy consumption and improved packet delivery ratio in comparison to SinkTrail.

Keywords: Aggregation; Data gathering; Mobile sink; Routing; WSNs.

1. Introduction

WSNs consist of sensor nodes that have limited energy, processing capacity, bandwidth and memory. Sensor nodes sense the physical environment and send their data to the sink. A number of activities are performed at each sensor node which includes sensing, computation, transmission, reception, sleep and wake up. Among all activities, transmission consumes the highest energy. The data from the sensor is normally transmitted through multi-hop communication. Data aggregation can significantly reduce the energy consumption by combining the data and thereby reducing the number of packets transmitted.Data aggregation can be push-based or pull-based. In push-based data aggregation, the sensor nodes push their data after aggregation to the sink.

----

* Corresponding author.

Push-based aggregation can be further categorized as periodic data aggregation in which data is gathered and aggregated at regular intervals of time or event-driven data aggregation where data is gathered and aggregated on occurrence of an event. On the other hand, in pull-based data aggregation, the user of the sensor network issues query to pull aggregated data from the WSN.

Mobile sinks were introduced in recent past to reduce the energy consumption. A mobile sink may be the human being carrying a Personal Digital Assistant (PDA), a laptop mounted on a vehicle or an unmanned aerial vehicle. The mobile sink moves around the sensor network to collect the data. This reduces the average path-length of the messages and hence the overall energy consumption of the WSNs. Many algorithms were proposed for scheduling the mobile sink for collecting data from WSN, most of which consider the task of scheduling the mobile sink for data collection as a traveling salesman problem. In [1] a dynamic data gathering protocol, SinkTrail is presented but its performance deteriorates with increase in size of the network.

1.1. Motivation

Most of the algorithms for data gathering using a mobile sink select few nodes as rendezvous nodes that collect data from the remaining nodes in the WSN. The energy of the rendezvous nodes drain off soon because during each data gathering round, the rendezvous nodes receive data from its neighbors and send it to the mobile sink. If a rendezvous node fails, the algorithm must be executed again to select a new set of rendezvous nodes and hence incurs a delay. It is challenging to design an algorithm that allows the mobile sink to collect data from the WSN irrespective of it's the moving pattern and node failures. In addition most of the data gathering protocols proposed for WSNs with mobile sink are well suited for periodic data collection.

1.2. Contribution

In this paper, we propose GADMS an algorithm with following contributions:

- Allows the mobile sink to collect data from sensor irrespective of the moving pattern of the mobile sink
- Yields less energy consumption, higher packet delivery ratio and less delay since the average path length of the data packet is constant.

1.3. Organization

This paper is organized as follows:

Section II presents a brief review of related works. Section III describes the preliminaries. Section IV defines the problem and describes the system model. Section V presents the algorithm GADMS. Section VI discusses the simulation, results and performance analysis. Section VII contains the conclusions.

2. Literature survey

Wireless sensor networks have higher error rate and low throughput than optical networks [3,4]. Gatzianas and

Georgiadis, [5] presented distributed algorithm for routing data towards mobile sink while maximizing the lifetime of WSNs.

Yang and his colleagues [6] designed a swarm-intelligence-based protocol for data collection SIMPLE, in WSNs with mobile sinks. Swarm agent in SIMPLE computes residual energy of the node to make routing decisions and hence maximizes lifetime of the sensor node and of the network. It is robust and scales with multiple mobile sinks but incurs delay and energy for initial swarm agent packets during initial route setup.

Wang and his colleagues [7] derived an upper bound for improving the lifetime of a sensor network with mobile relays and constructed a joint mobility and routing algorithm to achieve this bound in large WSNs. It yields better lifetime, delay and packet delivery ratio in comparison to static sinks in WSNs.

Yun and his colleagues [8] presented a framework that can maximize the network lifetime in WSNs running delay tolerant applications with mobile sink. The problem of selecting a route for mobile sink is formulated as optimization problem, that is maximize the lifetime of the WSN subject to constraints of delay bound, node energy and flow conservation. The algorithm is centralized and does not find the halting point of sink so as to maximize the network lifetime.

Kim and his colleagues [9] designed an Intelligent Agent based Routing (IAR) protocol for acquiring data from WSN with mobile sink. IAR incurs less packet loss, signalling overhead and control overhead. The Delay and Energy consumption increases with increase in number of sources and mobile sinks.

Nakayama and his colleagues [10] proposed Set Packing Algorithm and Traveling salesman problem (SPAT) for fair and energy efficient data gathering in Wireless Sensor and Actuator Network with mobile sink.

Rasheed and his colleagues [11] described a three-tier framework that uses two separate key pools one for authenticating the mobile sink and another for establishing pairwise key between sensor nodes. It provides better resilience to mobile sink and stationary access node replication attack.

Konstantopoulose and his colleagues [12] designed a five phase protocol, MobiCluster for data gathering and aggregation from sensor networks with a mobile sink. The Mobi-Cluster removes spatially and temporally correlated data and maximizes connectivity and data throughput while reducing the network overhead and balancing the energy consumption.

Shi and his colleagues [13] designed Random Line Walk (RLW) for forwarding messages in sensor network where nodes do not know the geographical location of each other apriori. Further, they proposed data discovery mechanism, *double cross* that exploits the geometric property of planar and uses RLW to forward messages. Double Cross improved data delivery rate and lower path length at a comparable average energy consumption.

Liu and his colleagues [14] developed a comprehensive theoretical Mobility Assisted Data Collection (MADC) model to obtain the boundary of the throughput capacity and network lifetime in WSNs with mobile sink. The performance analysis reveals that the MADC gives better performance than WSNs with static sink with respect to throughput and lifetime.

Ren and his colleagues [15] presented a distributed clustering algorithm for mobile sensor networks to group the nodes in the network and two distributed data aggregation algorithms, Distance-AGG and Probability-AGG that aggregate the data based on distance and probability respectively for aggregating data from different clusters in mobile sensor networks. Both data aggregation algorithms can deliver accurate result while reducing transmission cost and delay.

Danpu and his colleagues [16] have designed an energy-efficient transmission scheme that combines cluster-based MIMO technology and multi-hop communication, MIHOP. It incorporates an algorithm to compute the optimal number of hops needed to form the multi-hop network for transmission while remaining sensor nodes transmit using MIMO scheme. MIHOP shows better energy-efficiency in comparison to individual multi-hop scheme and virtual MIMO scheme.

He and his colleagues [17] formulated the problem of scheduling mobile nodes as Traveling Salesman Problem with Neighborhoods (TSPN) and based on which they proposed two algorithms Combine Skip Substitute(CSS) and multirate CSS(MR-CSS) schemes for scheduling the Mobile Element (ME) in WSNs with fixed communication range and variable communication range respectively. While using MR-CSS scheme, there is a possibility that the algorithm returns a shorter communication range when the Mobile Element uses a larger communication range and hence the available range is not utilized completely.

Liu and his colleagues [1] presented two proactive energy-efficient protocols, *SinkTrail* and *SinkTrail-S*, for gathering data from sensor networks with mobile sinks. SinkTrail allows data collection where the mobile sink does not follow a predefined path and uses logical coordinates for routing and forwarding decisions. SinkTrail is less complex and incurs low control overhead. There exists a trade-off between the frequency of data gathering and energy.

Li and his colleagues [18] proposed a lightweight routing structure update scheme with a mobile sink that constructs and updates the data collection tree by utilizing the special correlation. It incurs low data collection delay.

Thriveni and his colleagues [19] presented probabilistic-average energy-flooding (PAEF) that performs an averaging algorithm Calculate-Average-Energy (CAE) to make routing decisions at regular intervals and significantly reduces the energy consumption.

Kanavalli and his colleagues [20] proposed a flat routing protocol that considers residual energy of nodes to make routing decisions and thereby increases the network lifetime. Chandrakant and his colleagues [21] presented Energy efficient MIDdleware service (EMID) for maximizing lifetime of WSN by applying optimal sleep-wakeup policies.

Tarannum and his colleagues presented two routing protocols [22,23], for energy efficient routing in WSNs. The [22] Sink Administered Load Balanced Dynamic Hierarchical Protocol (SLDHP) minimizes the energy consumption of sensor nodes by balancing the load on cluster heads in a hierarchically organized sensor network, whereas in [23] Energy Efficient Routing Protocol (EERP) is presented that uses a modified version of directed diffusion technique to reduce the energy consumption.

Manjula and his colleagues [24] studied the impact of mobility models on the performance of Ad Hoc On-Demand Distance Vector Routing Protocol (AODV) routing protocol.

Prakash and his colleagues [25] designed Residual Energy Adaptive Re-Routing (REAR) scheme that solves the problem of maximizing the network lifetime as Mixed Integer Linear Programming (MILP).

Vibha and his colleagues [26] presented a framework for detecting congestions, and then predict the traffic flow.

3. Background

Liu and his colleagues [1] presented SinkTrail and its improvement SinkTrail-S protocols for data gathering in WSNs where the topology of the network is not known *apriori*. SinkTrail does not require assistance from any GPS device or predefined landmarks to incorporate terrestrial changes in the routing decisions. It allows sink to take any path for gathering data from the senor nodes at reduced energy consumption. SinkTrail uses logical coordinates for data gathering in WSNs with mobile sinks. Sink moves inside the WSN and halts at certain locations called trail points for gathering data from the sensor nodes. *SinkTrial* algorithm works in three phases: 1) Logical coordinate space construction 2) Destination identification and 3) Greedy Forwarding.

3.1. Logical coordinate space construction

SinkTrail uses *trail reference* to represent the logical coordinates. The *trial reference* is a d_v-dimensional vector that represents the distance of the sensor node from the mobile sink. The rightmost value in the trail reference indicates the distance of sensor node to current trail point of mobile sink where it halts for data collection and values towards left side of the trail reference are distance to previous d_v trail points in the reverse chronological order.

The mobile sink broadcasts its trail message with message sequence number and hop count value 0. Each sensor node stores the sequence number of the most recent message received, λ. When a sensor node X receives the trail message, it first compares the message sequence number with λ to see if it is a recent message. If it is a recent trail message not received before or already received message with shorter distance, then the node updates the hop count field in the message by 1 and rebroadcasts the message. Finally, it updates the rightmost value in the trail reference as message hop count + 1.

3.2. Destination Identification

The specialty of the logical coordinate is that irrespective of the halting point of the mobile sink the trail reference of sink is always [d_v-1, ..., 1, 0]. This is because each time sink moves to a new trail point, its hop count is set to 0 and previous trail point is 1 hop away and so on. The trail reference of mobile is called

Destination Reference that represents distance of the mobile sink's current trail point with that of previous d_v trail points.

3.3. Greedy Forwarding

Once the sensor node updates its trail reference, it sets a timer inversely proportional to the rightmost value in the trail reference, i.e., its distance in hop count to the current trail point of the mobile sink to allow the farthest node to transmit first. Each sensor stores a routing table consisting of the trail references of its neighbors. The sensor node computes the Euclidean distance of its trail reference with each of its neighbor and selects the one with smallest distance.

A property of the greedy forwarding in SinkTrail is that since the forwarding neighbor is chosen based on the trail reference, it does not require frequent change in the forwarding neighbor, especially for the nodes that are at far distance from the sink and hence results in energy savings. But since the distance is computed using trail reference the data delivery rate deteriorates.

4. Problem statement and system model

4.1. Problem statement

The sensor nodes are constrained in battery power and hence saving energy of the sensor network is a primary concern. Mobile sink based data gathering protocols are designed to save energy in WSNs and finds a widespread usage in surveillance applications. Many protocols are designed that schedule the mobile sink for gathering data from Sensor Networks. Most of them consider the problem of scheduling the mobile sink as traveling salesman problem and require information about the topology of the sensor network. The scheduling algorithms elect a few nodes as data gathering nodes or rendezvous nodes that deliver the data collected from its neighbors to the mobile sink.

A prior knowledge about the network may not be possible in all situations. In addition, the rendezvous points will become hot spots and suffer from energy depletion at a faster rate. Since the scheduling decision happens at a centralized location, failure of a rendezvous node requires rescheduling and incurs additional delay. An alternative design is to allow the mobile sink to take any path for collecting data from sensor network.

In the Background section SinkTrail [1] a dynamic data collection protocol is discussed that uses trail reference to make routing decisions. In SinkTrail each node keeps different trail reference corresponding to each mobile sink and hence as the number of mobile sink increases, there is an increase in energy consumption. In addition, there is a decrease in the packet delivery ratio as the distance between the sink and a sensor node increases.

Given a sensor network with N sensor nodes and M mobile sinks, the problem is to design algorithm that allow the sensor nodes to route their aggregated data to the mobile sink when path of the mobile sink is not predetermined in energy efficient manner while maintaining a higher data delivery rate and lower delay.

Objective

- Reduce the overall energy consumption
- Increase the packet delivery ratio when path of the mobile sink is not predetermined
- Reduce the average path length and thereby increase fault tolerance

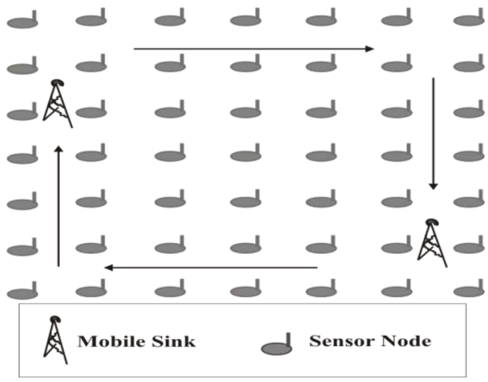

Figure1: Network deployment

4.2. System Model and Assumptions

4.2.1. Network model

The WSN consists of *N* sensor nodes placed in random manner across the network and *M* mobile sinks where $M \geq 1$ as shown in Fig.1. At regular intervals, the mobile sink moves around the sensor network to collect data. The path of the mobile sink is not predetermined and hence the mobile sink can take any path and halt at any point to gather data. The halting time of the mobile sink at each point and the number of halting points are large enough to collect data from the WSN. It is assumed that the mobile sink has sufficient battery power and storage space.

4.2.2. Communication model

All sensor nodes in the network are assumed to be connected. The communication range of each sensor node in the WSN is a constant say r. During every round, each sensor that has data to be transmitted generates a fixed size data packet and transmits it to the mobile sink using multihop communication. The Gathering Aggregated Data using Mobile Sink (GADMS) uses *HopLimit* field (η) that restricts the number of hops a message travels. The *HopLimit* field is used to ensure that a node that is more than η hops away from the mobile sink does not participate in the data gathering process at the current halting point of mobile sink. The *HopLimit* can effectively reduce the average path length of data packet and thereby results in increased packet delivery ratio at lower energy consumption. The *HopLimit* field is proportional to the network size.

4.2.3. Sink mobility model

The sink moves around the sensor network along a rectangular path, because the concept of *HopLimit* works well with the rectangular path in comparison to a random path or a diagonal path. The energy saving of rectangular movement is much better than that of other type of movements. At each halting point, the sink broadcasts query message and waits for a predetermined amount of time to gather data. We assume that the number of halting time is sufficiently large enough to gather partially aggregated data from the WSN. In WSNs with two mobile sinks, the sinks are assumed to move in opposite direction to gather data as shown in Fig. 1. Both the sinks propagate same query with same $query_{id}$. A node chooses the mobile sink with smaller *HopCount* as its destination. The movement pattern chosen in such a way that a node does not receive a query message from both sinks at the same time except in some exceptional conditions. The distance between first halting point and second halting point is chosen to be η-1.

5. Gathering Aggregated Data using Mobile Sink in Wireless Sensor Networks (GADMS)

The Gathering Aggregated Data using Mobile Sink (GADMS) protocol is designed to allow the sensor nodes send their partially aggregated data to the mobile sink. The GADMS algorithm is given in Algorithm 1. The mobile sink takes a rectangular path around the WSN to gather data. The rectangular path is chosen after analysis of the moving pattern in random, diagonal and rectangular pattern. Based on our analysis and discussions in [1], it is inferred that rectangular movement pattern yields higher energy savings in comparison to diagonal and random movement.

Table 1: List of Notations

Notation	Meaning
A_X	Aggregated data at node *X*
Id	Identifier of Sensor Node
X, *Y*	Sensor node with id *X* or *Y*
M	Query Message
q	Recent query id at a sensor node
Q	Query Id in message *M*
ξ	Sequence number of recent query message M received at a sensor node
HopCount	Hop count of a sensor node from the mobile sink
C	No of hops the Query message has traveled
HopLimit	Hop limit of recent query message *M* received at a sensor node
η	Hop limit of query message *M* received at a sensor node
DFlag	Flag to indicate whether data corresponding to *Q* is transmitted or not
DTimer	Timer for data transmission
T	Dwell time of Mobile sink at a halting
FNeighbor	Forwarding Neighbor of a sensor node

In case of a sensor network with two mobile sinks, the two sinks moves in opposite direction to gather data as shown in Fig. 1. This movement pattern of mobile sink ensures that a node does not receive a query message from two sinks at the same time. Once the data for a particular query is communicated to one sink, the data for the same query will not be sent to other sink. The GADMS algorithm works in three phases 1) Query Dissemination 2) Query propagation and route establishment and Data aggregation and forwarding. The three phases are discussed below:

5.1. Query Dissemination from the Mobile Sink

The procedure for query broadcast from mobile sink is depicted in the Algorithm 1. The mobile sink moves around the network to gather data. The Phase 1 of GADMS is executed when the mobile sink broadcasts query *M*. The query message *M* consists of the five tuple:

< Q, Query, ξ, C, η >

Where *Q* is the query id, ξ is the sequence number, *C* is the hop count and η is the the hop limit. Each sensor node stores four fields sent along with the query, in its local memory.

Function: BcastQry()

Create Query Message *M*

Initialize SeqNo to 0,

Set hop limit field η

foreach *HaltingPoint* **do**

  Increment SeqNo by 1

Set HopCount to 0

Broadcast query message M

Set Timer to *T*

Wait for incoming data packet

if *(Timer expired)* **then**

Move to next *HaltingPoint*

Figure 6: Function to Broadcast Query

During its tour, the mobile sink halts in some location for a small amount of time, *T* for gathering data. The value of halting time *T* is predetermined and is chosen to be large enough to acquire data from the sensor network. The mobile sink sets a timer with value *T* at each halting point and starts moving towards next halting point when the timer expires. At each halting point, the mobile sink broadcasts the same query message *M* with a new sequence number. The hop count field *C* in *M* indicate the number of hops the node is away from Sink. Hence the *C* field in the query message is set to 0 at the mobile sink. A hop limit η field decides the number of hops, the query message should survive and is used to ensure better data delivery ratio.

5.2. Query Propagation and Route establishment

When a node *X* receives query *id Q* from a node *Y*, where *Y* can be a mobile sink or an intermediary node, it first compares the query id of received message with *q*. If the query *id* of the received message is higher than *q*, indicates that this is new query which has not been received before. The node *X* stores the query *id* and sequence

Begin

Phase I: Query Broadcast by Mobile Sink

Call BcastQry()

Phase II: Query Propagation and Route Establishment

if *(Query message is received at Sensor* X*)* **then**

if *(*q < Q*)* **then**

Store Q, η *in X*

Set *HopCount* to *C*+1

Set X as FNeighbor

Set DFlag to FALSE

Reset DTimer for data transmission

if *(* *HopCount* < η*)* **then**

Increment *C* field in *M* by 1

Replace *Id* field in *M* by *X*

Broadcast query message *M*

else if *(q* = Q*)* **then**

if *(ξ = SeqNo)* **then**

if *(HopCount* > *C+1)* **then**

Set FNeighbor to *Id* field in *M*

Set *Id* field in *M* to *X*

Increment *C* in *M* by 1

Broadcast query message *M*

Phase III: Data Aggregation and Forwarding

if *(Node* X *has data to satisfy the Query)* **then**

Store sensed data in A_X

else

Set A_X to 0

if *(Data packet is received from neighbor)* **then**

Aggregate the received data with A_X

if *(Dtimer is Expired)* **then**

Send A_X to FNeighbor

Set DFlag to TRUE

End

Figure 7: GADMS: Gathering Aggregated Data using Mobile Sink

number fields from the query *Q* locally in the variables *q* and *ξ* respectively. Node *X* resets its the distance to the mobile sink locally in variable *HopCount* to *C*+1. If the HopCount is equal to the *HopLimit* field *η* in the query then the query message is dropped to ensure that the query message does not travel beyond the designated number of hops. If hop count is less than *η*, then the message is rebroadcast after incrementing the *C* by 1 and changing the sender *id*. The node *X* stores *Y* as its forwarding neighbor, *FNeighbor*. In addition, the node *X* sets timer for data forwarding.

If a message with same query *id* as q and same sequence number as ξ is received from another neighbor say Z, then the node X compares the C field of the received packet with *HopCount*. If C+1 is less than *HopCount*, i.e., new route is shorter, then the node X stores Z as its forwarding neighbor and updates its hop count field to C+1. It then compares if it is a border node that has same hop count as the hop limit field η. It rebroadcasts the query message with updated information if it is not a border node. If the query *id* q in node X is greater than that of query id in the received message, then the packet is dropped.

5.3. Data aggregation and forwarding

Once the sensor receives a query, it checks whether it has a data to satisfy the query. Let A_X represent the aggregated data at a node that must be forwarded. If a node X has data for transmission, the node first stores its reading in the A_X and initializes A_X to 0. When a node receives data from its neighbor corresponding to query Q, it performs aggregation. Once the data transmission timer *DTimer* fires, the node X forwards its data to the *FNeighbor*, which is selected during route establishment phase.

The Algorithm 1 describes pull based aggregation. If it is to be implemented on a WSN where the query is preloaded, then the broadcast message has only three fields, i.e., $< \xi, C, \eta >$

Thus it is easy to adapt the algorithm for push-based and pull-based data aggregation.

6. Results and Analysis

Simulations are conducted in NS2 simulator and c++ [27,28] performance of GADMS is compared with the state-of-art algorithm SinkTrail [1]. One or more mobile sinks move around the network in rectangular path to collect aggregated data from the WSN where the sensor nodes are randomly deployed. The mobile sink travels along a rectangular path because it yields better energy advantage in comparison to circular and random movements. In order to analyze the scalability of GADMS, the network size is varied from 100 nodes to 500 nodes. The metrics used for performance analysis include 1) Energy 2) Packet Delivery Ratio 3) Path Length and 4) Delay. The performance of GADMS is analyzed in WSN with one mobile sink and two mobile sinks. In case of

WSN with two mobile sinks, the distance between the two sinks is at least $(2*\eta)-1$.

6.1. Impact of Network Size on Average Energy Consumption

Fig. 2 shows the impact of network size on energy consumption per node. The mobile sink moves at a distance of η nodes from the boundary of the network where η is the hop-limit used in the GADMS algorithm. This ensures that even though multihop communication is used, the number of hops for query message and data is limited by η. The sensor nodes that are η hops away from the sensor network send their data to their neighbors selected using the GADMS algorithm. The intermediary nodes along the path aggregate the incoming data with its own generated data before transmitting. The hop-limit (η) and data aggregation at intermediary nodes leads to reduction in the overall energy consumption per node in the GADMS protocol when compared with SinkTrail.

The analysis has been performed with one mobile sink and two mobile sinks and the results are depicted in Fig. 2. In SinkTrail, a sensor node may receive configuration message from both the mobile sinks and maintains different trail reference for each mobile sink and thus average energy dissipated per node increases with the number of mobile sinks. In the GADMS protocol, the average energy consumption per node does not vary with increase in mobile sinks. This is because the path of the two mobile sinks is chosen in such a way that the two mobile sinks are far enough so that each sensor node lies within the *hoplimit* of only one of the mobile sinks. GADMS shows 65% improvement in energy consumption.

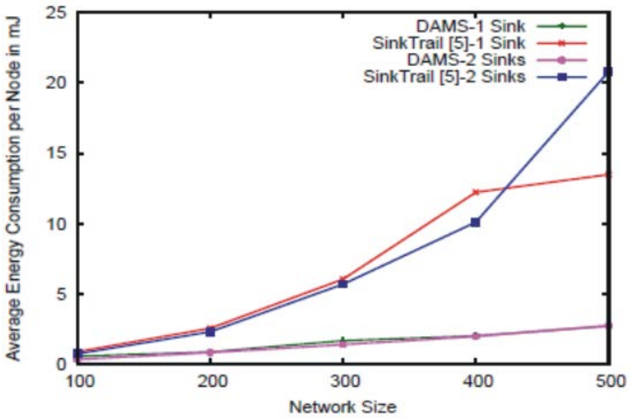

Figure 2: Energy vs. Network Size

6.2. Impact of Network Size on Average Packet Delivery Ratio

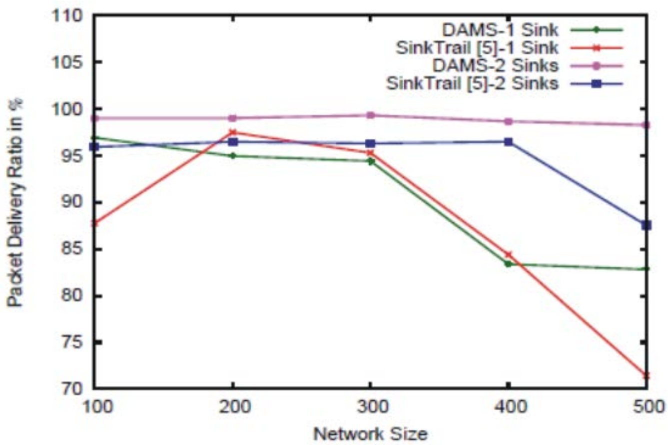

Figure 3: Average Packet Delivery Ratio vs. Network Size

Fig. 3 shows the impact of network size on packet delivery ratio (PDR) in WSN with one mobile sink and two mobile sinks. In WSNs with one mobile sink, the PDR decreases with increase in network size. This is due to the fact that the value of *hoplimit* increases with network size. Fig. 3 shows that the average packet delivery ratio of GADMS is better than that of SinkTrail.

The average PDR in GADMS is a constant in case of WSNs with two mobile sinks. On contrary in SinkTrail the average PDR decreases as the network size increases. This performance advantage of GADMS protocol is due to the use of *hoplimit* field. The *hoplimit* for any specific network size is a constant. Both sinks move in opposite direction gathering data. Thus, at any point of time, a sensor node comes within the *hoplimit* of only one sink. If it has already sent its data for a particular query, then the node does not send the same data again to another sink for query. Thus, GADMS results in improved PDR and energy advantage. GADMS with two sinks shows 7% improvement in PDR when compared to SinkTRail for larger networks.

6.3. Impact of Network Size on Average Path Length

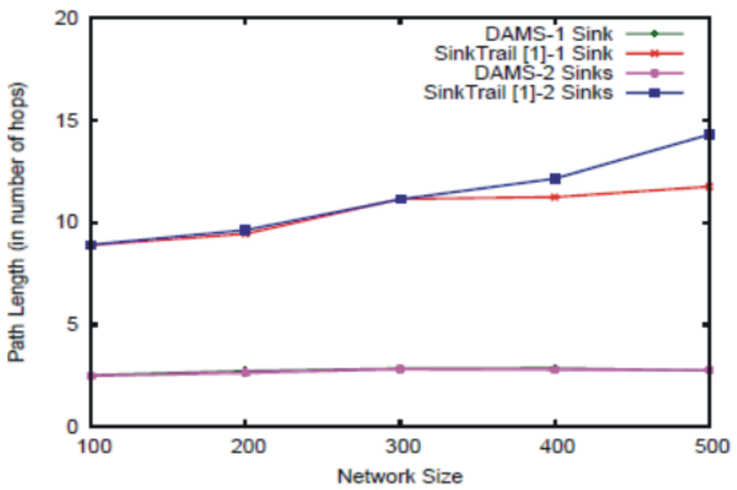

Figure 4: Average Path Length vs. Network Size

The average path length is measured as the number of hops the packet has to travel before the data reaches the mobile sink. Fig. 4 shows the average path length for different network sizes in case of sensor network with both one mobile sink and two mobile sink. It can be seen that in both the cases, the average path length of SinkTrail increases with the network size. On the other hand, in GADMS, the average path length is a constant irrespective of the network size and number of mobile sinks. This is because in GADMS, the sensor nodes aggregate all the received data before forwarding it to the neighbor and the *hoplimit* field ensures that any packet does not travel beyond η hops.

6.4. Impact of Network Size on Delay

Fig. 5. shows the impact of network size on delay in WSNs with one mobile sink and two mobile sinks. To maintain uniformity we have assumed that delay is the time between a query message sent by mobile sink to the time last data packet is received at the mobile sink in a halting point. Fig. 5 shows the average delay in WSNs with both one mobile sink and two mobile sinks. It can be seen that GADMS incurs a comparable delay with that of SinkTrail. GADMS incurs a delay slightly more than SinkTrail because in SinkTrail the sink halts at a location until enough data is collected.

In GADMS, the *Hoplimit* field is used that restricts the number of hops a message travels and hence incurs slightly more delay than that of SinkTrail. But this deterioration in delay contributes to improved performance in terms of energy, packet delivery ratio and path length.

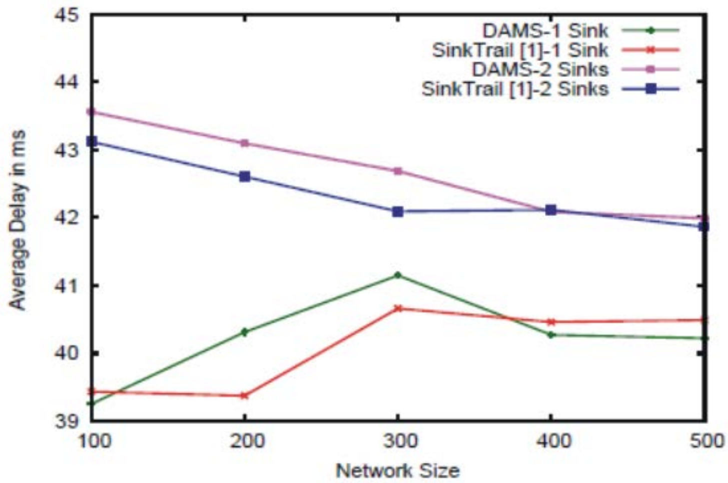

Figure 5: Delay vs. Network Size

7. Conclusions

We propose GADMS protocol that allows the mobile sink to acquire data from WSNs in which mobile sink does not follow a predetermined path. The mobile sink halts at a point in the network and broadcasts an aggregate query. The query is not flooded throughout the network, but to a limited number of hops. The data is collected from the limited number of hops and aggregated. This ensures uniform energy depletion among all nodes and thereby increases the network lifetime. Since the mobile sink dynamic and collects data from a limited number of nodes, the packet delivery ratio is improved. This also makes GADMS fault tolerant. The performance analysis shows that GADMS incurs low energy dissipation per node and improved packet delivery ratio at a comparable delay. As a future work we would like to improve GADMS by authenticating the mobile sink.

References

[1] Liu, H. Zhao, X. Yang, And X. Li, “Sinktrail: A Proactive Data Reporting Protocol For Wireless Sensor Networks,” IEEE Transactions On Computers, Vol. 62, No. 1, Pp. 151–162, 2013.

[2] F. Zhao And L. J. Guibas, “Wireless Sensor Networks: An Information Processing Approach,” Morgan Kaufmann, 2004.

[3] K. R. Venugopal, E. E. Rajan, and P. S. Kumar, “Impact of Wavelength Converters in Wavelength Routed All-Optical Networks,” Computer communications, vol. 22, no. 3, pp. 244–257, 1999.

[4] K. R. Venugopal, E. E. Rajan, and P. S. Kumar, “Performance Analysis of Wavelength Converters in WDMWavelength Routed Optical Networks,” 5th International Conference On High Performance Computing, HIPC’98, 1998, pp. 239–246.

[5] M. Gatzianas and L. Georgiadis, “A Distributed Algorithm for Maximum Lifetime Routing in Sensor Networks with Mobile Sink,” IEEE Transactions on Wireless Communications, vol. 7, no. 3, pp. 984–

994, 2008.

[6] H. Yang, F. Ye, and B. Sikdar, “A Swarm-Intelligence-Based Protocol for Data Acquisition in Networks with Mobile Sinks,” IEEE Transactions on Mobile Computing, vol. 7, no. 8, pp. 931–945, 2008.

[7] W. Wang, V. Srinivasan, and K. Chua, “Extending the Lifetime of Wireless Sensor Networks Through Mobile Relays,” IEEE/ACM Transactions on Networking (TON), vol. 16, no. 5, pp. 1108–1120, 2008.

[8] Y. Yun and Y. Xia, “Maximizing the Lifetime of Wireless Sensor Networks with Mobile Sink in Delay-Tolerant Applications,” IEEE Transactions on Mobile Computing, vol. 9, no. 9, pp. 1308–1318, 2010.

[9] J.W. Kim, J. S. In, K. Hur, J. W. Kim, and D. S. Eom, “An Intelligent Agent-based Routing Structure for Mobile Sinks in WSNs,” IEEE Transactions on Consumer Electronics, vol. 56, no. 4, pp. 2310–2316, 2010.

[10] H. Nakayama, Z. M. Fadlullah, N. Ansari, and N. Kato, “A Novel Scheme for WSAN Sink Mobility Based on Clustering and Set Packing Techniques,” IEEE Transactions on Automatic Control, vol. 56, no. 10, pp. 2381–2389, 2011.

[11] Rasheed and R. N. Mahapatra, “The Three-Tier Security Scheme in Wireless Sensor Networks with Mobile Sinks,” IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 5, pp. 958–965, 2012.

[12] Konstantopoulos, G. Pantziou, D. Gavalas, A. Mpitziopoulos, and B. Mamalis, “A Rendezvous-Based Approach Enabling Energy-Efficient Sensory Data Collection with Mobile Sinks,” IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 5, pp. 809–817, 2012.

[13] G. Shi, J. Zheng, J. Yang, and Z. Zhao, “Double-Blind Data Discovery using Double Cross for Large-Scale Wireless Sensor Networks With Mobile Sinks,” IEEE Transactions on Vehicular Technology, vol. 61, no. 5, pp. 2294–2304, 2012.

[14] W. Liu, K. Lu, J. Wang, G. Xing, and L. Huang, “Performance Analysis of Wireless Sensor Networks with Mobile Sinks,” IEEE Transactions on Vehicular Technology, vol. 61, no. 6, pp. 2777–2788, 2012.

[15] Q. Ren, L. Guo, J. Zhu, M. Ren, and J. Zhu, “Distributed Aggregation Algorithms for Mobile Sensor Networks with Group Mobility Model,” Tsinghua Science and Technology, vol. 17, no. 5, pp. 512–520, 2012.

[16] L. Danpu, Z. Kailin, and D. Jie, “Energy-Efficient Transmission Scheme for Mobile Data Gathering in Wireless Sensor Networks,” China Communications, vol. 10, no. 3, pp. 114–123, 2013.

[17] L. He, J. Pan, and J. Xu, “A Progressive Approach to Reducing Data Collection Latency in Wireless Sensor Networks with Mobile Elements,” IEEE Transactions on Mobile Computing, vol. 12, no. 7, pp. 1308–1320, 2013.

[18] Z. Li, Y. Liu, M. Li, J. Wang, and Z. Cao, “Exploiting Ubiquitous Data Collection for Mobile Users in Wireless Sensor Networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 2, pp. 312–326, 2013.

[19] J. P. Thriveni, K. R. Vishwanath, K. R. Venugopal, and L. M. Patnaik, “Probabilistic Average Energy Flooding to Maximize Lifetime of Mobile Ad Hoc Networks,” Third International Conference on Wireless Communication and Sensor Networks, WCSN’07, 2007, pp. 65–68.

[20] A. Kanavalli, D. Serubiri, P. D. Shenoy, K. R. Venugopal, and L. M. Patnaik, “A Flat Routing Protocol for Sensor Networks,” In Proceedings of International Conference on Methods and Models in Computer Science, ICM2CS, 2009, pp. 1–5.

[21] N. Chandrakant, J. Tejas, D. Harsha, P. D. Shenoy, K. R. Venugopal, and L. M. Patnaik, “EMID: Maximizing Lifetime of Wireless Sensor Network by Using Energy Efficient Middleware Service,” International Journal of Information and Electronics Engineering, vol. 2, no. 3, pp. 314–317, 2012.

[22] S. Tarannum, S. Srividya, D. S. Asha, R. Padmini, L. Nalini, K. R. Venugopal, and L. M. Patnaik, “Dynamic Hierarchical Communication Paradigm for Wireless Sensor Networks: A Centralized, Energy Efficient Approach,” In Proceedings of 11th IEEE Singapore International Conference on Communication Systems, ICCS, 2008, pp. 959–963.

[23] S. Tarannum, B. Aravinda, L. Nalini, K. R. Venugopal, and L. M. Patnaik, “Routing Protocol for Lifetime Maximization of Wireless Sensor Networks,” In Proceedings of International Conference on Advanced Computing and Communications, 2006, pp. 401–406.

[24] S. H. Manjula, C. N. Abhilash, K. Shaila, K. R. Venugopal, and L. M. Patnaik, “Performance of AODV Routing Protocol using Group and Entity Mobility Models in Wireless Sensor Networks,” In Proceedings of the International MultiConference of Engineers and Computer Scientists, vol. 2, 2008.

[25] T. S. Prakash, G. S. Badrinath, K. R. Venugopal, and L. M. Patnaik, “Energy Aware Topology Management in Ad Hoc Wireless Networks,” In Proceedings of International Conference on Systems and Networks Communications, ICSNC’06, 2006, pp. 1–12.

[26] L. Vibha, M. Venkatesha, P. G. Rao, S. N, P. D. Shenoy, K. R. Venugopal, and L. M. Patnaik, “Moving Vehicle Identification using Background Registration Technique for Traffic Surveillance,” Proceedings of the International MultiConference of Engineers and Computer Scientists, 2008, pp. 572–577.

[27] K. R. Venugopal, K. G. Srinivasa, and L. M. Patnaik, Soft Computing for Data Mining Applications. Springer, 2009.

[28] K. R. Venugopal and R. Buyya, Mastering C++. Tata McGraw-Hill Education, 2013, ISBN(13): 978-1-25902994-3, ISBN(10): 1-25-902994-8.