DATA GATHERING IN WIRELESS SENSOR NETWORKS WITH UNCONTROLLED SINK MOBILITY

*R.Dinesh kumar, Assistant *Professor, Department of Computer Science ,*
Bharathiyar college of Engineering and Technology, Karaikal
*S.Flora**, II year M.Tech computer science,*
Bharathiyar college of Engineering and Technology, Karaikal

ABSTRACT

In some applications, the moving path of the mobile sink is unpredictable. For sensors that have detected an event of interest, it is necessary to find the location of the mobile sink first before they can report the data. In this paper, we will discuss how source sensors can locate the constantly moving mobile sinks and send data packets through a less number of forwarding hop counts. The effectiveness of the proposed algorithm is verified by comparing with other data gathering via uncontrolled mobile sink algorithms. Moreover, our experimental results also confirm that the proposed algorithm can ensure that all source sensors can find the location of the mobile sink and reduce the average forwarding hop counts of data packets.

Keywords- *wireless sensor networks, static sensor, uncontrolled mobile sink, data gathering*

1.INTRODUCTION

A Wireless Sensor Network (WSN) is a network system composed of numerous wireless sensors and a few sinks which transmit data wirelessly. WSN are commonly applied in areas such as military, agriculture, disaster prevention and emergency rescue. In all these applications, sensors gather environmental information and then report the sensing data to the sink(s) wirelessly. In some applications, the mobile sink may move freely. So, the moving path of the mobile sink is unpredictable. For example, intelligence reporting in the battlefield is an application of this operation. A group of soldiers with a handheld PDAs patrolling in battlefield to gathering information from the WSN. To send the sensing data to the mobile sink, the source sensor has to locate the mobile sink first and then deliver the data via multi-hop transmission. Hence, how to improve the packet delivery ratio is a primary focus of this type of research. For example, intelligence reporting in the battlefield is an application of this operation. A group of soldiers with a handheld PDAs patrolling in battlefield to gathering information from the WSN. To send the sensing data to the mobile sink, the source sensor has to locate the mobile sink first and then deliver the data via multi-hop transmission. Hence, how to improve the packet delivery ratio is a primary focus of this type of research.

In order to enhance the packet delivery ratio, Shi et al. [7][8] proposed the Data-Driven Routing Protocol (DDRP). In DDRP, sensors can be divided into three types, including One- Hop neighboring Sensor nodes of mobile sinks (OHS), Multi- Hop neighboring Sensor nodes of mobile sinks (MHS) and Infinite-Hop neighboring Sensor nodes of mobile sinks (IHS). overhearing-based route learning process, sensors without beacon packets are able to overhear the route information when their neighboring sensors with route information deliver data packets. The status of sensors which have overheard the route information will be changed from IHS to MHS. Because OHS sensors and MHS sensors possess route information, they can deliver data packets to mobile sinks by tracing beacon packets. IHS sensors do not have route information, so they need to rely on random walk to find OHS/MHS sensors. Hence, DDRP still cannot guarantee 100% delivery of data packets. that is, they cannot ensure that all data packets can be delivered to a mobile sink. Besides, as data packets are delivered by random walk and following the beacon packets, the number of times of data relay will be very high.

In this paper, we attempt to improve the above-mentioned problem. The proposed algorithms are expected to ensure 100% delivery of sensing data and also reduce energy consumption by delivering the data through a smaller number of times of data relay compared to existing algorithms

2. LITERETURE SURVAY

2.1 “Data gathering protocols for wireless sensor networks with mobile sinks,”

Wireless sensor networks with mobile sinks (mWSN) have attracted a lot of attention recently. In an mWSN, each mobile sink can move freely and unpredictably. In this paper, we design two efficient data gathering protocols for mWSNs. The first protocol (called AVRP) adopts Voronoi scoping plus dynamic anchor selection to handle the sink mobility issue. In the second protocol (called TRAIL), the trail of mobile sink is used for guiding packet forwarding as sinks move in the network. In TRAIL, to forward a data packet, integration of trail-based forwarding and random walk is used. Specifically, when no fresh trail of any sink is known, random walk is used; once a sensor on a fresh sink trail is reached, data packet will be forwarded along the trail. TRAIL is simple to implement and has small protocol overhead. Simulation results show the designed protocols have high performance and further AVRP is suitable for mWSNs with heavy traffic while TRAIL is suitable for mWSNs with light traffic.

2.2. “A progressive approach to reducing data collection latency in wireless sensor networks with mobile elements,”

The introduction of mobile elements has created a new dimension to reduce and balance the energy consumption in wireless sensor networks. However, data collection latency may become higher due to the relatively slow travel speed of mobile elements. Thus, the scheduling of mobile elements, i.e., how they traverse through the sensing field and when they collect data from which sensor, is of ultimate importance and has attracted increasing attention from the research community. Formulated as the traveling salesman problem with neighborhoods (TSPN) and due to its NP-hardness, so far only approximation and heuristic algorithms have appeared in the literature, but the former only have theoretical value now due to their large approximation factors. In this paper, following a progressive optimization approach, we first propose a combine-skip-substitute (CSS) scheme, which is shown to be able to obtain solutions within a small range of the lower bound of the optimal solution. We then take the realistic multi rate features of wireless communications into account, which have been ignored by most existing work, to further reduce the data collection latency with the multi rate CSS (MR-CSS) scheme. Besides the correctness proof and performance analysis of the proposed schemes, we also show their efficiency and potentials for further extensions through extensive simulation

2.3 “Dependable wireless sensor networks for reliable and secure humanitarian relief applications,”

Disasters such as flooding, earthquake, famine and terrorist attacks might occur any time anywhere without prior warnings. In most cases it is difficult to predict when a disaster might occur however, well-planned disaster recovery procedures will reduce the intensity of expected consequences. When a disaster occurs, infrastructure based communications are most likely to be crippled, worsening the critical situation on hand. Wireless ad hoc and sensor network (WASN) technologies are proven to be valuable in coordinating and managing rescue operations during disasters. However, the increasing reliance on WASNs make them attractive to malicious attackers, especially terrorist groups, in a bid to hamper rescue operations amplifying the damage and increasing the number of casualties. Therefore, it is necessary to ensure the fidelity of data traffic through WASN against malicious traffic disruption attacks. In this paper, we first demonstrate how WASN can be used in a well-planned disaster recovery effort. Then, we introduce and analyze one of the most severe traffic disruption attacks against WASNs, called *Identity Delegation*, and its countermeasures. Its severity lies in its capability to evade detection by even state-of-the-art intrusion detection techniques such as the neighbor monitoring based mechanisms. Through identity delegation, an adversary can drop packets, evade detection, and frame innocent nodes for dropping the traffic. We introduce a technique to mitigate identity delegation attack, dubbed SADEC, and compare it with the state-of-the-art mitigation technique namely Basic Local Monitoring (BLM) under a wide range of network scenarios.

2.4 “Energy-efficient routing protocols in wireless sensor networks: a survey,”

The distributed nature and dynamic topology of Wireless Sensor Networks (WSNs) introduces very special requirements in routing protocols that should be met. The most important feature of a routing protocol, in order to be efficient for WSNs, is the energy consumption and the extension of the network's lifetime. During the recent years, many energy efficient routing protocols have been proposed for WSNs. In this paper, energy efficient routing protocols are classified into four main schemes: Network Structure, Communication Model, Topology Based and Reliable Routing. The routing protocols belonging to the first category can be further classified as flat or hierarchical. The routing protocols belonging to the second category can be further classified as Query-based or Coherent and non-coherent-based or Negotiation-based. The routing protocols belonging to the third category can be further classified as Location-based or Mobile Agent-based. The routing protocols belonging to the fourth category can be further classified as QoS-based or Multipath-based. Then, an analytical survey on energy efficient routing protocols for WSNs is provided. In this paper, the classification initially proposed by Al-Karaki, is expanded, in order to enhance all the proposed papers since 2004 and to better describe which issues/operations in each protocol illustrate/enhance the energy-efficiency issues.

3.PROPOSED WORK

To conserve the energy of the sensor devices since only some nodes, called cluster heads (CHs), are allowed to communicate with the base station.

The representative design is low-energy adaptive clustering hierarchy (LEACH) protocol which uses a pure probabilistic model to select CHs and rotates the CHs periodically in order to balance energy consumption.

The main motivation is to utilize distributed clustering for scalability, to employ mobility for energy saving and uniform energy consumption, and to exploit Multi-User Multiple-Input and Multiple-Output (MU-MIMO) technique for concurrent data uploading to shorten latency. The main contributions of this work can be summarized as follows.

First, we propose a distributed algorithm to organize sensors into clusters, where each cluster has multiple cluster heads.

Second, multiple cluster heads within a cluster can collaborate with each other to perform energy efficient inter-cluster transmissions.

Third, we deploy a mobile collector with two antennas (called SenCar in this paper) to allow concurrent uploading from two cluster heads by using MU-MIMO communication. The SenCar collects data from the cluster heads by visiting each cluster. It chooses the stop locations inside each cluster and determines the sequence to visit them, such that data collection can be done in minimum time.

The purpose of LEACH-C is to produce better clusters by dispersing the cluster head nodes throughout the network.

3.1 NETWORK ENVIRONMENT AND PROBLEM FORMULATION

This paper considered a scenario in which a set N of static sensors was distributed over a plane of dimension $l \times w$, where $N=\{n_i \mid 1\ \delta i\ \delta n,\ n=|N|\}$ and n_i is a static sensor. We assume the sensor network is connected. Moreover, we assume that there is a mobile sink in the field. The mobile sink can move freely and unpredictably. The communication range of the mobile sink and static sensors is r_c. An example of WSN is shown in Fig. 1.As mentioned above, we hope to achieve 100% delivery of data packets to the mobile sink. So we set the first goal of this problem as expressed in Equation (1), where D is a set of all data packets in the network ($D=\{d_i \mid 1\ \delta i\ \delta d,\ d=|D|\}$), d_i is a data packet, and $R(d_i)\{0, 1\}$. If data packet d_i is received by the mobile sink, $R(d_i)=1$; otherwise, $R(d_i,)=0$.

$(\ )\ 1_{id}d_{\square}\angle\ =_{D}R(1)$

Compared to packets containing the footprint information, general data packets are larger in size. Hence, it is better that data packets are delivered to the mobile sink over a shorter distance (i.e. hop counts). The following (Equation (2)) presents the second goal of this problem, where $H(d_i)$ is the hop distance between the source sensor of data packet d_i to the mobile sink.

To reduce the overhead of finding the routing path to the mobile sink, we have to reduce the traffic flow of packets generated by searching the location of the mobile sink. The following (Equation 3) presents the third goal of this problem, where F is the traffic flow generated by the location information of the mobile sink. $T(d_i)$ is the traffic flow generated by sending data packet d_i to the mobile sink. $Q(d_i)$ is the traffic flow generated by searching for the routing path for data packet d_i.

3.2 THE CONCEPT AND APPROACH

As mentioned earlier, both TRAIL and DDRP algorithms find the mobile sink by tracing beacon packets. A beacon packet contains the following information: sink ID and timestamp (beacon packet = $\langle ID_b, t_b \rangle$. However, they fail to provide the guarantee because they cannot ensure that all data packets can reach a sensor with the beacon packet. In this section, Trail-based Algorithm with Guide line (TAG) is proposed for sensor to proactively report their data back to the mobile sink. In TAG, we set some virtual guide lines (Definition 1) as a means to ensure that all source sensors can deliver their data packets to the mobile sink. When a mobile sink passes through a virtual guide line, it will send a guide packet (Definition 2) to a sensor closest to it. This sensor will then become the initial guide-line sensor (Definition 3). Next, the initial guide-line sensor will send guide packets to the sensor above it and the sensor below it. The two sensors will relay the guide packet in the same direction to the next sensor above/below it. Sensors which have received a guide packet will also become guide-line sensors. This operation continues until further relay of the guide packet in the delivery direction is not possible. Under the assistance of guide-line sensors, when any sensor has data packet to send to a mobile sink, it only needs to pass the data packet in horizontal directions, and the data packet can be delivered a mobile sink in the network. Definition 1: virtual guide line GL_v is a set of virtual guide lines, where $GL_v=\{x_1, x_2, \dots, x_v\}$.Take Fig. 2 as an example. $GL_v=\{0, 50, 100\}$ is a set of 3 virtual guide lines. The linear equations of the five virtual guide lines are x=0, x=50 and x=100. When the mobile sink moves to any location where the x-coordinate is 0, 50 or 100, it will send a guide packet to a sensor closest to it. Definition 2: guide packet A guide packet contains a sequence number (seq_g), the sink ID (ID_g), timestamp (t_g) and the number of hop counts of relay (*counts*), where guide packet = ⟨seq_g, ID_g, t_g, *counts*⟩. Each sensor will maintain a Beacon Table, but only guide-line sensors have the Guide Table. Definition 3: guide-line sensor Sensors having the guide packet are defined as guide-line sensors. Guide-line sensors whose guide line packet comes from a mobile sink are called initial guide-line sensors. In the following subsection, we will explain how we develop TAG based on the above-mentioned concept to ensure 100% delivery of data packets to the mobile sink.

3.2.1. Maintaining the Guide Table

When the sensor receives a guide packet directly from a mobile sink, it will store the guide packet in its Guide Table and resend it to sensors above and below it. Take Fig. 2 as an example. Suppose that n_1 receives a guide packet ⟨s231000, A, 034508100000, 0⟩ from the mobile sink. n_1 will store the guide packet in its Guide Table. Because this guide packet comes from a mobile sink, n_1 is an initial guide-line sensor. When n_2 and n_5 receive the guide packet relayed from n_1, both n_2 and n_5 will add 1 to *counts* of the guide packet and store this guide packet ⟨s231000, A, 034508100000, 1⟩ in the Guide Table .Later, they will relay the packet to n_3 and n_6 respectively. After receiving the guide packet, n_3 and n_6 will also add 1 to *counts* of the guide packet and store the packet ⟨s231000, A, 034508100000, 2⟩ in the Guide Table. The guide packet will be further relayed to n_4 and n_7, both of which will add 1 to *counts* of the guide packet after receiving it and store the packet ⟨s231000, A, 034508100000, 3⟩ in the Guide Table. By this time, for n_4 and n_7, there is no more sensor to relay the guide packet. Hence, the relay operation will be terminated. In this example, sensors n_1~n_7 have the guide packet, so they are guide-line sensors.

3.2.2. The relay of guide packets.

A. Finding the Mobile Sink

When a sensor detects an event of interest, it will pack the event information into a data packet and send query packets (Definition 4) to sensors on its left and right sides to find the mobile sink. Definition 4: query packet A query packet contains the initial sender of the query message (n_i), a sequence of sensor identifications (*list*), the sequence number of the guide packet (seq_g), the timestamp of the guide packet (t_g), the guide-line sensor of the guide packet (n_g), the receiver of the beacon packet (n_b) and the timestamp of the beacon packet (t_b), where query packet = ⟨n_i, *list*, (seq_g, t_g, n_g), (n_b, t_b) ⟩.

Take the illustration in Fig. 3 as an example to explain how to find the mobile sink. When n_s detects an event of interest, it will pack the event information into a data packet ⟨n_s, *event*⟩ and send query packet ⟨n_s, *list*, ⎞, ⎞⟩ to the sensor on its left (n_8) and the sensor on its right (n_{12}) to find the mobile sink. We explain how the sensors on the left side of n_s assist in relaying query packets. When n_8 receives a query packet from n_s, it will add its id to the *list* in the query packet and then update (n_b, t_b) and (seq_g, t_g, n_g) in the query packet based on its Beacon Table and Guide Table. As n_8 is not located along the footprint of mobile sink A, its Beacon Table will not contain any record of mobile sink A. Besides, mobile sink A passes through virtual guide line 3 at timestamp = 034510220000, and n_8 is the guideline sensor of this virtual guide line. Hence, there is a record of mobile sink A ⟨s231002, A, 034510220000, 2⟩ in n_8's Guide Table. Sensor n_8 will then update the query packet into ⟨n_s, n_8, (s231002, 034510220000, n_8), ⎞⟩ and send this query packet to n_9 along the forwarding direction.

B. An example of transmitting query/route/data packets

When n_9 receives a query packet from n_8, it will add its id to the *list* in the query packet and then update (n_b, t_b) and (seq_g, t_g, n_g) in the query packet based on its Beacon Table and Guide Table. Because n_9 is neither located along the footprint of mobile sink A nor located on the virtual guide line, it does not have any record in its Guide Table and Beacon Table. This means n_9 will not update (n_b, t_b) and (seq_g, t_g) fields in the query packet. In other words, the query packet will be updated to ⟨n_s, (n_8, n_9), (s231002, 034510220000, n_8), ⎞⟩ and send this query packet to n_{10} along the forwarding direction.

When n_{10} receives a query packet from n_9, it will add its id to the *list* in the query packet and then update (n_b, t_b) and (seq_g,t_g, n_g) in the query packet based on its Beacon Table and Guide Table. Sensor n_{10} is also not on the virtual guide line, so it does not have any record of the mobile sink in its Guide Table. Because the mobile sink has passed by n_{10}, given that the passage occurs at timestamp = 034509500000, n_{10} has a beacon packet ⟨A, 034509500000⟩ in its Beacon Table. In other words, the query packet will be updated to ⟨n_s, (n_8, n_9, n_{10}), (s231002, 034510220000, n_8), (n_{10}, 034509500000)⟩ and send this query packet to n_{11} along the forwarding direction.

When n_{11} receives a query packet from n_{10}, it will add its id to the *list* in the query packet and then update (n_b, t_b) and (seq_g, t_g, n_g) in the query packet based on its Beacon Table and Guide Table. As n_{11} is not located along the footprint of mobile sink A, its Beacon Table will not contain any record of mobile sink A. Besides, mobile sink A passes through virtual guide line 2 at timestamp = 034509150000, and n_{11} is the guide-line sensor of this virtual guide line. Hence, there is a record of mobile sink A ⟨s123457, A, 034509150000, 1⟩ in n_{11}'s Guide Table. However, the timestamp on this record is older than the record in the query packet from n_{10}, this record will not be used to update the (seq_g, t_g) field in the query packet. In other words, the query packet will be updated to $\langle n_s, (n_8, n_9, n_{10}, n_{11}), (s231002, 034510220000, n_8), (n_{10}, 034509500000)\rangle$ by n_{11}.During transmission of the query packet, query packets are forwarded from virtual guide lines of newer timestamp to virtual guide lines of older timestamp. It is not possible to find a newer virtual guide line if the packet is continuously forwarded. Hence, n_{11} will stop sending the query packet and return the routing path it has obtained. In this instance, from record in the (seq_g, t_g, n_g) and (n_b, t_b) fields in the query packet, we can find that the guide packet owned by n_8 is newer than the beacon packet owned by n_{10}. Therefore, n_{11} will set n_s as the source of the route packet and n_8 as the destination of the route packet and then plan the routing path based on record in the *list* field. Specifically, n_{11} will return a route packet of $\langle n_s, n_8, 034510220000, n_s \rightarrow n_8\rangle$. Similarly, sensors on the right of n_s, including n_{12}, n_{13}, n_{14}, and n_{15}, will find the last footprint of the mobile sink in the same way**.**

C. Determining the Routing Path of Data Packets

If the sensor (which sends the query packet) receives the route packet from sensors on its left and right, it will extract the guide information in its Guide Table and the beacon information in its Beacon Table and find the latest route information. Later, it will send the data packet based on the latest routine information. As shown in Fig. 3, n_{12} has a new time stamp of the mobile sink, so n_s will choose $n_s \rightarrow n_{12}$ as the routing path. When a sensor receives a data packet where it is the destination in the *list* field, it will follow the footprint (i.e. beacon packet) of the mobile sink to deliver the data packet to the mobile sink. As shown in Fig. 3, after the data packet is delivered to n_{12}, n_{12} will follow the guide packet to send the data packet to n_{16}. Subsequently, n_{16} will also follow the guide packet to send the data packet to n_{17}. n_{17} will follow the beacon packet to send the data packet to n_{18}, n_{18} will follow the beacon packet to send the data packet to n_{19}. When n_{19} receives the data packet, it will deliver the data packet to the mobile sink.

Target field (two-dimensional plane $l \times w$) (500~1500) × 500 units of distance2 Number of sensors (sensor density: 0.5%) 1250~3750Communication range of the mobile sink/sensor 20 units of distance Communication range of sensor 20 units of distance Velocity of the mobile sink 1 unit of distance / unit of time A round 10 units of time Event probability 10% sensors per round Beacon interval 20 units of time Interval of virtual guide line 50 units of distance Sensing data size 512 bytes Beacon packet size 10 bytes Guide packet size 13 bytes

4.SIMULATION RESULT

We evaluate the performance of the proposed TAG with two other algorithms, including, TRAIL [1] and DDRP [7][8].We evaluate the performance of these algorithms in terms of the delivery ratio of data packets, the average forwarding hop counts of data packets and overall network traffic. The simulation parameters are listed in Table 1. We built our simulator in java programs. For each experiment, we performed 1000 units of time to obtain the mean of the results.

A. Delivery Ratio of Data Packets

First, we compare the delivery ratio of data packets with different value of TTL between the three algorithms in an area of 1000 · 500 units of distance$_2$. The comparison of the delivery ratio across different value of TTL is shown in Fig. 4.As shown in Fig. 4, with the increase in the value of TTL, the delivery ratio by these algorithms also increases. Among these algorithms, the delivery ratio of the TRAIL algorithm is lowest. We find that even given a high TTL (i.e. TTL=1000), TRAIL and DDRP cannot guarantee a 100% delivery ratio. As shown in Fig. 4, given TTL=1000, the delivery ratio by TRAIL and DDRP algorithms are 41% and 68% respectively. These two algorithms' inability to guarantee a 100% delivery ratio is associated with their use of random walk in scenarios where the sink's location information is unavailable. Compared to TRAIL and DDRP algorithms, the proposed TAG algorithm can reach 100% delivery ratio if TTL is greater than or equal to 100. The superior performance is mainly contributed by the virtual guide lines.

B. Average Forwarding Hop Counts of Data Packets

Next, we compare the average forwarding hop counts of data packets between the three algorithms in networks of different sizes. To obtain the average number of forwarding hop counts of data packets, we set TTL=1000 and calculate the average among data packets that have been successfully delivered to the mobile sink. Fig. 5 shows a comparison of the average forwarding hop counts of data packets between the three algorithms. For instance, given an area of 1000 · 500 units of distance$_2$, the average forwarding hop counts of data packets by TRAIL, DDRP and TAG algorithms are 648, 183 and 61 respectively. The comparison among the three algorithms shows that the proposed TAG requires an obviously smaller number of average forwarding hop counts. Moreover, as shown in this figure, with the increase of the network size, the average forwarding hop counts of data packets also increases. There is only one mobile sink in the region. In a larger region, the average distance between sensors to the mobile sink will also be larger. In this case, more forwarding hop counts are needed for source sensors to deliver their data packets to the mobile sink.

C. Overall network traffic and Message Overhead of Finding the Routing Path of Data Packets

Compared to TRAIL and DDRP algorithms, TAG has the additional overhead of guide packets, query packets, and route packets. In the following experiment, we will estimate the amount of data generated by control messages (i.e. beacon packets + guide packets + query packets + route packets). The query packet size equals to 18 bytes + *list* · 2bytes and the route packet size equals to 12 bytes + *list* · 2bytes. Fig. 6 shows the message overhead of finding the routing path of data packets by TAG algorithm. As shown in Fig. 6, with the increase of network size, the message overhead of find the routing path of data packets by TAG also increase. This is because the larger the network, the harder it is for source sensors to find the mobile sink. A larger hop distance for sending query packets and route packets is required to find the mobile sink.

Although TAG has the additional overhead of guide packets, query packets, and route packets, they generate a smaller overall network traffic as compared to TRAIL and DDRP algorithms. In other words, the additional guide packets, query packets, and route packets transmitted by TAG are helpful for reducing the overall network traffic. Fig. 7 shows a comparison of the overall network traffic. As shown in this figure, the overall network traffic generated by TAG algorithm is far smaller than that by TRAIL and DDRP algorithms across various network sizes. For instance, the overall network traffic generate by TRAIL, DDRP and TAG algorithms in an area of $1000 \cdot 500$ units of distance$_2$ are 7908 MB, 2232 MB and 1209 MB respectively. The results shown in Fig. 7 suggest that TAG can reduce the overall network traffic. It should be especially noted that TAG can ensure 100% delivery ratio for all source sensors given TTL=100 (see Fig. 4). In the same condition, TRAIL and DDRP algorithms can reach a delivery ratio of 12% and 41% respectively. This finding is very important in some data gathering applications.

5. CONCLUSION

In this paper, we revisit the data gathering via uncontrolled sink mobility problem. The proposed TAG algorithm can guarantee 100% delivery of data packets to the mobile sink. Through simulation experiments, we confirm that the proposed TAG algorithm outperform the existing data gathering algorithms for uncontrolled sink mobility (TRAIL [1] and DDRP [7][8]) in terms of delivery ratio of data packets, the average forwarding hop counts of data packets, and overall network traffic.

REFERENCES

[1] K. Tian, B. Zhang, K. Huang, J. Ma, “Data gathering protocols for wireless sensor networks with mobile sinks,” *Proc. IEEE GLOBECOM’10*, pp. 1-6, 2010.
[2] L. He, J. Pan, J. Xu, “A progressive approach to reducing data collection latency in wireless sensor networks with mobile elements,” *IEEE Trans. Mobile Comput.*, vol. 12, no. 7, pp. 1308-1320, 2013.
[3] I.M. Khalil, A. Khreishah, F. Ahmed, K. Shuaib, “Dependable wireless sensor networks for reliable and secure humanitarian relief applications,” *Ad Hoc Netw.*, vol. 13, pp. 94-106, 2014.
[4] N.A. Pantazis, S.A. Nikolidakis, D.D. Vergados, “Energy-efficient routing protocols in wireless sensor networks: a survey,” *IEEE Commun.Surveys Tuts.*, vol. 15, no. 2, pp. 551-591, 2013.
[5] M.V. Ramesh, A. Shanmughan, R. Prabha, “Context aware ad hoc network for mitigation of crowd disasters,” *Ad Hoc Netw.*, vol. 18, pp. 55-70, 2014.
[6] T. Rault, A. Bouabdallah, Y. Challal, “Energy efficiency in wireless sensor networks: A top down survey,” *Comp. Netw.*, vol. 67, no. 4, pp. 104-122, 2014.
[7] L. Shi, B. Zhang, K. Huang, J. Ma, “An efficient data-driven routing protocol for wireless sensor networks with mobile sinks,” *Proc. IEEE ICC’11*, pp. 1-5, 2011.
[8] L. Shi, B. Zhang, H.T. Mouftah, J. Ma, “DDRP: An efficient data-driven routing protocol for wireless sensor networks with mobile sinks,” *Int. J. Commun. Syst.*, vol. 26, pp. 1341-1355, 2013.
[9] H. Wang, H.E. Roman, L. Yuan, Y. Huang, R. Wang, “Connectivity, coverage and power consumption in large-scale wireless sensor networks,” *Comp. Netw.*, vol. 75, pp. 212-225, 2014.
[10] X. Yu, P. Wu, W. Han, Z. Zhang, “A survey on wireless sensor network infrastructure for agriculture,” *Comput. Stand. Inter.*, vol. 36, no. 1, pp. 134-146, 2013.
[11] M. Yun, D. Bragg, A. Arora, H.A. Choi, “Battle event detection using sensor networks and distributed query processing,” *Proc. IEEE INFOCOM WKSHPS’11*, pp. 750-755, 2011.