A New Distributed Routing Protocol for Wireless Sensor Networks with Mobile Sinks

Hengyi Wen[1], Zheng Yao[1], Huiqiang Lian[2,3], and Baoxian Zhang[1](✉)

[1] Research Center of Ubiquitous Sensor Networks,
University of Chinese Academy of Sciences, 19A Yuquan Road, Beijing 100049, China
wenhengyi15@mails.ucas.ac.cn, {yaozheng,bxzhang}@ucas.ac.cn

[2] University of Chinese Academy of Sciences, 19A Yuquan Road, Beijing 100049, China
lianhuiqiang15@mails.ucas.ac.cn

[3] Information Center, PetroChina Hebei Company, Shijiazhuang 050000, Hebei, China

Abstract. Mobile sinks have been widely used for data gathering in mobile-enabled wireless sensor networks (mWSNs). However, the frequent network topology dynamics caused by sink mobility in mWSNs may cause excessive protocol overhead for route discovery and maintenance and therefore routing performance degradation. In this paper, we design a new distributed routing protocol for achieving high routing performance in such networks. The designed routing protocol works in a reactive way for route learning/updating and it combines trail based forwarding and data-driven route learning/updating for improved performance as sink(s) move in the network. We present detailed design description of the new protocol. Simulation results show that our protocol can achieve high packet delivery ratio performance with low protocol overhead.

Keywords: Routing protocol · Wireless sensor networks · Mobile sinks

1 Introduction

Recently, wireless sensor networks with mobile sinks have been an attracting paradigm for data gathering in target environment. A wireless sensor network with one or multiple mobile sinks (MS) is typically referred to as mWSN. Much existing work has shown that use of mobile sinks can largely improve the performance of a WSN as compared with use of static sinks. However, sink mobility can cause unpredictable changes in network topology, which brings great challenges to the design of efficient routing protocols for such networks wherein sensor nodes only have limited resources and capabilities. Therefore, how to design efficient routing protocols for mWSNs to handle such sink mobility while achieving high routing performance with low protocol overhead has been a key issue in the research and design of mWSNs.

This work was supported partially by the National Natural Science Foundation of China under grants 61471339, 61531006, 61173158.

C. Li and S. Mao (Eds.): WiCON 2017, LNICST 230, pp. 150–159, 2018.
https://doi.org/10.1007/978-3-319-90802-1_13

Much work has been carried out in the area of routing in mWSNs and existing protocols in this area can be divided into the following three types [1–3]: Location based routing protocols, topology based routing protocols, and reactive routing protocols. Location based protocols (e.g., LURP [4], TTDD [5], and ER [6]) require nodes in the network to have their own locations, their neighbors' locations, and packet destinations' locations. In these protocols, the location information will be used for guiding geographical packet forwarding in a hop-by-hop manner. However, in many cases, obtaining accurate location information of nodes is not always practical, in particular for mWSNs wherein mobile sinks can move rapidly and unpredictably. Topology based protocols (e.g., AVRP [7] and MDRP [8]) typically requires nodes in the network to form an efficient routing structure for packet delivery from sensor nodes to nearby mobile sinks. Topology based protocols can typically acquire short paths for packet delivery at the excessive cost of protocol overhead for path maintenance, which is often not attractive in WSNs with sporadic traffic. Reactive routing protocols (e.g., TRAIL [7] and DDRP [9]) works reactively for path delivery and maintenance, which has demonstrated good routing performance while having very low protocol overhead. In this paper, we shall focus on design of reactive routing protocol for mWSNs.

In this paper, we design an efficient distributed (reactive) routing protocol for mWSNs. The design objective is to achieve high routing performance with low protocol overhead. To achieve this goal, our protocol integrates trail based forwarding, data-driven packet forwarding, and random walk routing. In this way, our protocol combines sink-mobility-driven route learning and neighbor-forwarding-driven route learning with small protocol overhead. In the use of these forwarding strategies, routing distance and route freshness as characterized by time stamp freshness are used in the next hop selection. When a sensor node has no valid route for reaching a mobile sink, however, random walk routing is enforced until the data packets reach a sensor node on a fresh trail or with a valid route to reach a mobile sink. We present detailed protocol design of the protocol. Simulation results show that our protocol can achieve high packet delivery ratio performance with low protocol overhead.

The rest of this paper is scheduled as follows. In Sect. 2, we briefly review some existing routing protocols for mWSNs. In Sect. 3, we present the design of the new routing protocol. In Sect. 4, we conduct simulations to evaluate the performance of the designed protocol by comparing it with existing work. In Sect. 5, we conclude this paper.

2 Related Work

Much work has been carried out to support efficient routing in mWSNs and existing protocols in this aspect can be divided into the following three types: Location based routing protocols, topology based routing protocols, and reactive routing protocols. All these routing protocols can enable sensor nodes in the network to report their sensed data to a nearby mobile sink via multi-hop routing in a real-time fashion. Next, we shall respectively introduce typical protocols belonging to each of these types.

2.1 Location Based Routing Protocols

Location based routing protocols utilize node location information for assisting geographical packet forwarding in an mWSN. The major advantages of location based routing include simplicity, good routing performance, and high scalability. Typical location based routing protocols for mWSNs include Local Update-Based Routing Protocol (LURP) [4], Two-Tier Data Dissemination (TTDD) [5], and Elastic Routing (ER) [6].

LURP [4] works to restrict the location updating scope of mobile sink to a small circle other than the entire network, whenever possible. According to LURP, mobile sink chooses a small circular area and issues its up-to-date location inside the circle (at a high frequency) as long as it is still inside the circle. However, when it moves outside the circle, network-wide location updating will be triggered and a new circle will be chosen for local location updating but with higher frequency. Data packets outside the circle are forwarded toward the circle via geographical forwarding while topology-based routing is used inside the circle.

TTDD [5] provides scalable and efficient data delivery from a data source to multiple mobile sinks. Each data source proactively builds a grid structure in the network by dividing the sensing field into cells with dissemination nodes located at the crossing points of the grid. The delivery structure by TTDD is easy to maintain. However, with the increase of data source number and sink number, TTDD can produce excessive protocol overhead.

In ER [6], source sensors node can keep obtaining the newest location information of a mobile sink by its continuous data reporting to the sink. More concisely, when a sink node moves, its new location information is propagated backward along the data path to the source sensor via piggybacking the freshest sink location information into each data packet for forwarding downstream. Such piggybacking can allow the upstream node on a path to learn the freshest sink location in a hop-by-hop manner. Continuous data packet delivery in the forward direction enables the source node to learn the up-to-date location of its communicating mobile sink.

2.2 Topology Based Routing Protocols

Topology based protocols in general work proactively for route discovery and maintenance and they typically build efficient routing structure in the network, which provides short routes from each sensor node in the network to a nearby mobile sink at the cost of protocol overhead. Example protocols in this type include AVRP [7] and MDRP [8].

AVRP [7] has good performance in mWSNs with heavy traffic and infrequent movement of mobile sinks. It can reduce the protocol overhead by use of Voronoi scoping and dynamic selection of anchor node for each mobile sink in order to hide the short movement of the sinks. AVRP assumes that there are multiple mobile sinks moving in the sensing field uncontrollably and each of them selects a neighbor sensor as its anchor node. Voronoi scoping is used for restricting each mobile sink's interest dissemination scope for route updating.

MDRP [8] is designed to improve the performance of AVRP. It divides the Voronoi scope associated with each mobile sink into multiple disjoint layers according to the distance of sensor nodes away from the anchor node associated with the mobile sink. When a mobile sink moves inside the first layer, route updating will be only made inside the first layer. When it leaves the first layer and enters the kth level, route updating will be made in layers $\leq k$. In this way, MDRP is expected to largely reduce the protocol overhead as compared with AVRP with little sacrifice in routing performance.

2.3 Reactive Routing Protocols

Reactive routing protocols work in a reactive way for route discovery and updating. Typical protocols in this category include TRAIL and DDRP. The route discovery and maintenance in TRAIL is triggered by sink mobility while that in DDRP is triggered by neighbors' packet forwarding. Next, we shall respectively introduce how either protocol works.

TRAIL [7] is a combination of random walk and trail-based packet forwarding. TRAIL takes advantage of the trail left by mobile sink as it moves in the network. When a sensor has data packet(s) to report, if it has (fresh) trail to reach any mobile sink, trail based forwarding will be used. This process continues until the packet reaches a mobile sink along the trail. However, when no such trail is known, random walk will be used until reaching a mobile sink directly, a fresh sink, or timed out and thus dropped without further processing.

DDRP [9] takes advantage of the broadcast feature of wireless medium for gratuitous route learning/updating and thus reduce the protocol overhead for data reporting. In DDRP, each data packet carries an additional option recording the known distance from the sender of the packet to a target mobile sink. Overhearing of such a packet transmission will gratuitously provide those listening neighbors a route to reach mobile sink. Continuous such route-learning among neighbor nodes will provide fresh routes to more and more sensor nodes in the network.

Our protocol in this paper shall integrate the trail based forwarding strategy in TRAIL and the data-driven route learning/updating strategy in DDRP for further improved routing performance.

3 Protocol Design

In this section, we shall present the detailed design process of our protocol, which combines "trail based forwarding and data-driven routing and is referred to as TBD. We will first give an overview regarding how TBD works, then give necessary routing information to be kept at sensor nodes for TBD to work correctly. Finally, we present the detailed design description of TBD.

3.1 Protocol Overview

To achieve improved routing performance in mWSNs, TBD inherits the data-driven route learning capability in DDRP and also the sink-mobility-driven route learning capability in TRAIL. It accordingly utilizes the following strategies for packet forwarding: trail based forwarding, data-driven routing, and random walk routing. Trail based forwarding is to take advantage of the trail information left by sink mobility as sinks move in the network. Moreover, to efficiently handle trail broken issue, TBD adopts two-hop local broadcast for trail repair with limited protocol overhead. Data-driven routing is to take advantage of the routing information learnt via overhearing of data packet transmissions at neighbor nodes. Random walk routing is triggered when a packet holder has no any routing information to reach a sink node. In TBD, when a sensor node has a data packet to forward to a sink node, trail-based forwarding has the highest priority, data-driven routing is the second, and random walk has the lowest priority.

Compared with DDRP and TRAIL, TBD enables more nodes in the network to learn fresh routing information by inheriting the data-driven route learning capability in DDRP and the sink-mobility-driven route learning capability in TRAIL. In this way, it can largely reduce the probability at which random walk has to be triggered and also shorten the routing distance and therefore improve the routing performance.

3.2 Routing Information

For TBD to work properly, we make the following assumption: All nodes in the network (including both sensors and sinks) are equipped with omnidirectional antennas and have the same communication range. Furthermore, no location information of nodes is known. Next, we will introduce necessary routing information to be kept at sensor nodes for TBD to work correctly.

Table 1. Routing information kept at sensor nodes.

Items	Description
TRAIL_flag	A bool flag variable, which indicates whether a sensor node is on a trail or not. If it is true, it means the sensor is on a fresh trail; otherwise, it is false
DDRP_flag	A bool flag variable indicating whether a sensor node has a valid route to reach a mobile sink as learnt via neighbor-forwarding operations
Time_stamp	It records the time when a route to mobile sink was generated and it was copied from the time stamp carried in a data packet containing fresh route to a mobile sink or a beacon message
HopDist	*HopDist* records the distance from the current node to a sink
NextHopID	*NextHopID* is the ID of next-hop to reach a sink
ExpireTime	*ExpireTime* is the time when the current entry will expire if no further update received

According to TBD, there are at most two entries to be kept at each sensor node in the network, one is for data forwarding and the other is for backup. The information in either entry is listed in Table 1. In Table 1, the first two items indicate in which way the routing information was learnt, time_stamp tells when the routing information was generated as stamped by the corresponding mobile sink. Initially, all nodes have no information about route to a sink and all the information in Table 1 will be Null.

Next, we introduce how the routing tables at sensor nodes are created and updated.

To enable trail based forwarding, the following procedure is taken for trail learning and updating. When a mobile sink moves in the network, it keeps broadcasting beacon messages periodically to its one-hop neighboring sensor nodes so as to leave a trail at sensors in the network. Accordingly, each neighboring sensor node of mobile sink knows that it can directly report data packets to a mobile sink. A beacon message carries the following information: The identifier of the mobile sink and a time stamp indicating when the message was generated. The sensor nodes which receive such beacon messages will locally create and update the corresponding routing tables as follows. It sets the flag TRAIL_flag = true, records the time stamp in its routing entry as that carried in the last received beacon message, and updates the ExpireTime field to Time_stamp + $\alpha \times T$ where α is a network parameter and T is the beacon interval.

To enable data-driven route learning/updating like in DDRP, each data packet contains an extra IP option Dist2mSink, which records the best known distance from the transmitter of the data packet to a nearby mobile sink. For example, if Dist2mSink equals to 3, then the transmitter of the packet is now three hops away from its target sink node. In actual protocol implementation, we assume Dist2mSink has a max value K. That is, a routing table only records routing information with distance $<K$ to reach a sink. There is a tradeoff between route learning scope and route freshness by adjusting the value of K. In the protocol operations, overhearing of such a packet transmission will gratuitously tell those listening neighbors fresh routes to reach a mobile sink. Continuous such route-learning among neighbor nodes will provide fresh routes for more and more sensor nodes in the network.

3.3 Data Packet Forwarding

In TBD, the priority of packet forwarding from the highest to lowest is as follows: trail based forwarding, data-driven routing/forwarding, and random walk routing. Accordingly, when fresh trail is available, trail based forwarding will be taken; else if data-driven learnt routing information is applicable, data-driven routing is performed; otherwise, random walk routing is taken. Figure 1 gives an example illustrating how a packet forwarding process works in the network when different routing information is available at different nodes. In this example, packet generated by a source sensor node is first forwarded via random walk, and then travel along route prepared by DDRP, and finally travel along a fresh trail before being reported to a mobile sink.

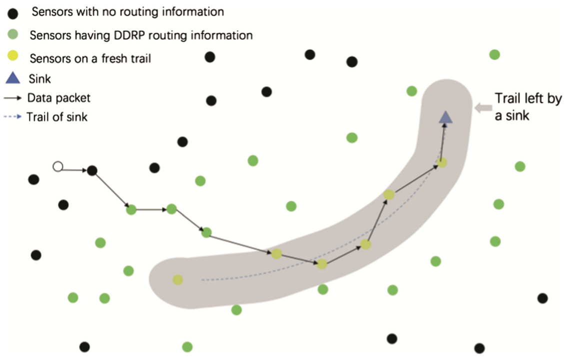

Sensors with no routing information
Sensors having DDRP routing information
Sensors on a fresh trail
Sink
Data packet
Trail of sink
Trail left by a sink

Fig. 1. An example illustrating how a packet is forwarded to a sink in TBD.

In TBD, once receiving a data packet from a neighbor sensor node or from the application layer directly, a sensor *u* will look up the routing entries on in its local cache, and carry out the following procedure based on which case the current packet holder is in:

Case 1: TRAIL_flag = true

In this case, we have sensor *u* is on a fresh trail. In this case, it will first look for sink(s) in its direct communication range. If it can find one such sink, the data packet will be forwarded directly to the mobile sink. If no sink can be found in its communication range, which means the sink has moved away, then trail based forwarding is triggered. In the trail based forwarding, the packet holder will send out a query message to see whether there is any neighbor with fresher sink-related record and start a timer. A query packet contains the identifier of the querying node and a time stamp copied from the local cache. If a reply is received before the timer expires, it will send the data packet to the sender of the reply packet. When multiple neighbors having fresher records than the packet holder, the one with the freshest record is expected to reply first via certain reply-deferring mechanism to suppress those unnecessary replies. If no reply is received, which means the trail is broken, then we use a two-hop local broadcast mechanism to find alternate route to bypass the broken point on the trail.

Specifically, the two-hop local broadcast mechanism for trail repair works as follows. The packet holder composes a new query message carrying its ID and time stamp information and then broadcasts this query to its one-hop neighbors. Upon receipt of this message, its one-hop neighbors will forward this message further. If any two-hop neighbor having fresher sink record than the packet holder, it will send a reply back to the packet holder. If multiple replies were received, the one with freshest record will be used. Upon receiving such a reply, the packet holder will send the packet to the node which sent it the reply message.

In case no reply is received in the trail repair process after certain time, sensor *u* will resort to its backup routing entry by going to Case 2. If no valid backup routing entry, random walk will be triggered by going to Case 3.

Case 2: DDRP_flag = true

In this case, sensor *u* has valid routing entry/entries learnt via overhearing of neighbor transmission(s). In this case, sensor *u* will perform data-driven routing by choosing the next hop from its routing entry and send the packet to the selected next hop.

Case 3: DDRP_flag = TRAIL_flag = false

In this case, node *u* does not have any valid routing information in its local cache, it will perform random walk routing by randomly selecting a neighbor sensor among its neighbor list and then send the packet to the selected neighbor.

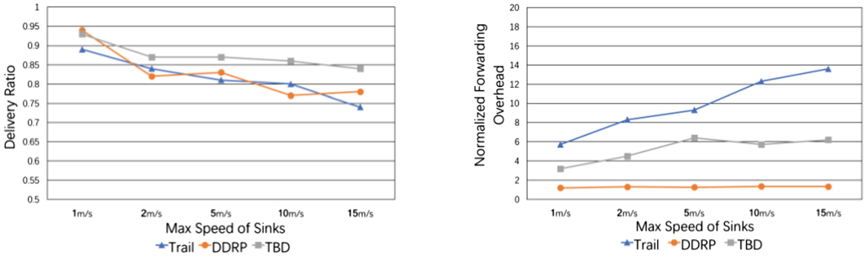
The sensor node receiving the data packet will repeat the above routing procedure until the packet reaches a mobile sink or timed out and then dropped.

4 Performance Evaluation

In this section, we conduct simulations for performance evaluation. The protocols simulated include the following: TBD in this paper, DDRP, and TRAIL. All these protocols do not require location information of nodes and they work reactively for path learning and updating. The simulation code was developed using Java. In the simulations, 200 sensor nodes and multiple mobile sinks were deployed uniformly at random initially in a 500 × 500 m^2 square area. Each node's communication range is 60 m. For each experiment, the simulation time is 1000 s.

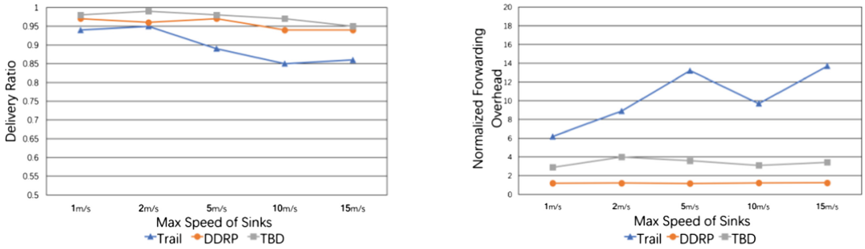
In our simulations, we used two metrics to evaluate the routing performance: Forwarding overhead and packet delivery ratio. The forwarding overhead is defined as the ratio of the total number of all packets transmitted (containing data packets and control packets) to the number of successfully delivered data packets. The number of packets transmitted is the total number of packet transmitted by all nodes (containing sensor nodes and sink nodes) and it also includes those transmissions of the packets that are forwarded, dropped, or collided. The lighter this overhead is, the more efficient a protocol will be. The packet delivery ratio is defined as the ratio of the total number of data packets successfully received by sinks to the number of data packets generated by sensor nodes. This metric represents the data delivery efficiency of a routing protocol.

In this experiment, the maximum velocity of mobile sinks varied from 1 to 15 m/s. The number of mobile sinks will be 1 and 3, respectively. Then, we will test different situations caused by the mobility of sink nodes. In the 3-sink case, we will observe routing performance when the packet generation rate in the network is 1 and 2 packets per seconds, respectively. Figure 2 shows the scenario with one sink and packet generation rate of 2 packets/s. Figure 2(a) shows the change of delivery ratio performance as the max speed of sinks change, Fig. 2(b) shows the change of overhead as the max speed of sinks change. Figure 3 shows the 3-sink situation. Figure 4 shows the 3-sink situation but changing the packet generation rate to one packet/s.



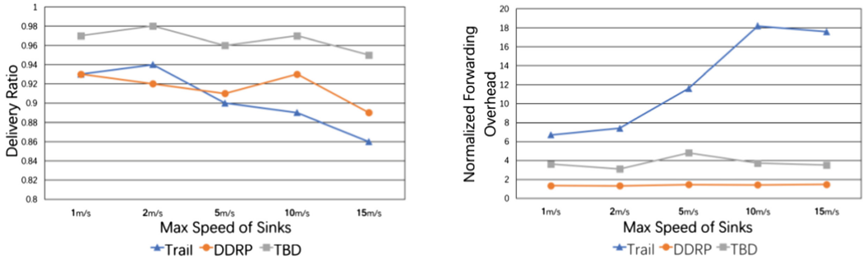
(a) (b)

Fig. 2. Performance of packet delivery ratio and normalized forwarding overhead versus sink speed. There is one sink in the network and the packet generation rate is two packets per second.



(a) (b)

Fig. 3. Performance of packet delivery ratio and normalized forwarding overhead versus sink speed. There are three sinks in the network and the packet generation rate is two packets per second.



(a) (b)

Fig. 4. Performance of packet delivery ratio and normalized forwarding overhead versus sink speed. There are three sinks in the network and the packet generation rate is one packet per second.

Through the above simulation results, we can see that TBD has the highest packet delivery ratio performance while having moderate routing protocol overhead as compared with DDRP and TRAIL. The higher protocol overhead by TBD as compared with DDRP is due to the overhead introduced for enabling trail based forwarding (i.e., exchanging of query-reply along trail and two-hop local broadcast for trail repair). In addition, it can be seen that sink number, sink moving speed, and packet generation rate have big impact on routing performance. Especially, in Figs. 2 and 3, we can see that as sink(s) move faster and faster, the packet delivery ratio keeps reducing while the overhead keeps increasing. Moreover, TBD has the highest packet delivery ratio performance as sink velocity increases. Figure 4 shows that, in lighter traffic situation, TBD gets even better performance on packet delivery ratio performance as compared with the other two protocols. Generally speaking, more sinks and lower moving speed lead to higher packet delivery ratio performance, and the faster sinks move, the more overhead will be caused.

5 Conclusion

In this paper, we have designed a distributed routing protocol for wireless sensor networks with mobile sinks, which integrates trail based forwarding, data-driven packet forwarding, and random walk routing. We present detailed design description of our protocol. Simulation results demonstrate that our protocol can improve the delivery ratio performance while keeping low protocol overhead.

References

1. Song, L.: Architecture of wireless sensor networks with mobile sinks: multiple access case. Int. J. Distrib. Sens. Netw. **3**(3), 289–310 (2007)
2. Liang, S., Hatzinakos, D.: Architecture of wireless sensor networks with mobile sinks: sparsely deployed sensors. IEEE Trans. Veh. Technol. **56**(4), 1826–1836 (2007)
3. Sheng, Y., Zhang, B., Li, C., Mouftah, H.: Routing protocols for wireless sensor networks with mobile sinks: a survey. IEEE Commun. Mag. **52**(7), 150–157 (2014)
4. Wang, G.J., Wang, T., Jia, W.J., Guo, M.Y. Chen, H.H., Guizani, M.: Local update-based routing protocol in wireless sensor networks with mobile sinks. In: Proceedings of the IEEE ICC 2007, pp. 3094–3099 (2007)
5. Ye, F., Luo, H.Y., Cheng, J., Lu, S.W., Zhang, L.X.: A two-tier data dissemination model for large-scale wireless sensor networks. In: Proceedings of the ACM MobiCom 2002, pp. 148–159 (2002)
6. Yu, F., et al.: Elastic routing: a novel geographic routing for mobile sinks in wireless sensor networks. IET Commun. **4**(6), 716–727 (2010)
7. Tian, K., Zhang, B., Huang, K., Ma, J.: Data gathering protocols for wireless sensor networks with mobile sinks. In: IEEE Global Telecommunications Conference (GLOBECOM 2010) (2010)
8. Shi, L., Yao, Z., Zhang, B., Li, C., Ma, J.: An efficient distributed routing protocol for wireless sensor networks with mobile sinks. Int. J. Commun. Syst. **28**(11), 1789–1804 (2015)
9. Shi, L., Zhang, B., Mouftah, H., Ma, J.: DDRP: an efficient data-driven routing protocol for wireless sensor networks with mobile sinks. Int. J. Commun. Syst. **26**(10), 1341–1355 (2013)