Mobile Data Collection Networks for Wireless Sensor Networks

Kai Li Kien A. Hua

Department of Electrical Engineering and Computer Science, University of Central Florida
Orlando, Florida 32816, U.S.A.
kailee.cs@knights.ucf.edu, kienhua@eecs.ucf.edu

Abstract. Energy consumption is a major limitation of wireless sensor networks, with irreplaceable tiny sensors and multi-hop communication in sensor data aggregation as major contributors to this problem. While existing single hop data collection schemes reduce individual sensor energy consumption, they suffer from longer latency and low data delivery ratio. In face of these problems, we propose a *mobile data collection network* (MDCNet) as a new paradigm for wireless sensing applications. MDCNet is a fully self-deployed mesh network with virtual mesh nodes (mobile relay nodes), through which sensor data can be collected in a single hop and transmitted to the sinks. Our simulation results, based on NS-2, indicate that this new approach can achieve short latency and high data delivery ratio.

Keywords: Wireless sensor network; mobile relay node; mobile data collection network

1 Introduction

Traditional wireless sensor networks (WSNs) are composed of densely-populated inexpensive tiny sensor nodes equipped with application-specific sensing units and communication components. The function of these sensors is to sample physical quantities (e.g. temperature, moisture, etc.) from the surroundings and route sensed data to a data processing center through base-stations (sinks). Since these tiny sensors have limited energy and they are often irreplaceable for many applications, energy consumption is of primary concern to ensure a long network operational time. Typically, data collection methods are based on sensors forming a connected network, through which data could be routed to the sink(s) through multiple hops. In this approach, a sensor is not only a data source, but also forwards data for other sensors. Such a data forwarding strategy may incur significant energy consumption. To reduce energy consumption and prolong the lifetime of WSNs, researchers have added mobility to WSN designs [7][8][9].

Mobility designs in WSNs falls into two major categories: **i) Using Mobile Sinks** [5][9][10]. Mobile sinks are used to replace static sinks as the endpoints of the data flow at the edge of the WSN. In general, sensors in the proximity of the stationary sinks will run out of energy faster because more data would pass through them to reach the sinks. This “energy hole problem” [6] results in premature cessation of network operation. Mobile sinks address this problem by moving to different locations in the field effectively balancing the data-forwarding workload for the sensor nodes. This data collection paradigm can extend network lifetime for many applications. **ii) Using Messenger Nodes** [1][4][8]. Sensor nodes do not relay data in this approach. Instead, intermediate mobile nodes act as messengers between the sensors and the stationary sink. These messenger nodes move around in the sensor field to collect sensed data from the sensors and deliver the data to the sink in a manner similar to delay tolerant networks [12]. As in the Mobile Sink approach, since sensor nodes do not need to relay data, the Messenger Node approach also substantially reduces the energy consumption for sensor nodes to prolong their service time.

Although the two aforementioned data collection strategies alleviate the energy consumption problem, they have some limitations. Data delivery through messenger nodes may incur excessive data collection delay. Considering the limited buffer space in tiny sensors, this may result in data loss if some sensors have to wait for a long time before their next chance to transmit data. Although using mobile sinks does not have this issue, they are relatively more expensive mobile devices with direct access to Internet and sometime powerful computing capability [11]. Deploying a large number of such mobile sinks to ensure short transmission delay for a large sensor field can be very expensive. This motivates us to consider a mobile data collection network (MDCNet) in this paper. An MDCNet consist of mobile relay nodes (MRN's) and one or more sink nodes. The MRNs form a network that allows electronic transmission of sensed data from the sensors to the sinks to minimize delay. MDCNet can be viewed as an advanced wireless mesh network [13] with the following two additional features: (1) unlike stationary mesh nodes which are deployed manually, our mobile relay nodes (MRNs) have the intelligence to survey the sensor field and automatically configure themselves into a mesh topology suitable for the given data collection task. (2) While mesh nodes are stationary, MRNs can move around in their designated region to achieve the effect of a virtual mesh node with a larger communication radius.

The advantage of the MDCNet approach is twofold. First, the degree of node mobility can be controlled (by deploying the proper number of MRN's) to achieve the desired tradeoff between transmission delay and overall system cost. This gives us a sensing system that is more energy efficient, shorter sensing latency, and higher data delivery ratio compared to existing mobile solutions. Second, since an MDCNet can be fully self-deployed, it is more suitable for inaccessible or hostile environments. In practice, the MRN's can be airborne (e.g., quadrocopter) or land-based (e.g., mini autonomous vehicle). This topic is beyond the scope of this paper. We simply refer to them as MRN's. To the best of our knowledge, MDCNet is the first data collection network designed for wireless sensing applications, and this paper is the first study of a fully self-deployed wireless mesh network with virtual mesh nodes.

The remainder of this paper is organized as follows. In Section 2, we briefly discuss some related work. Our MDCNet is introduced in Section 3. We present our simulation results in Section 4. Finally, we conclude this paper in Section 5.

2 Related Work

Optimization of sink movement for energy minimization is first studied in [2], in which the authors used integer linear programming to determine the locations for each mobile sink to stop by periodically for data collection. Stefano et al. [5] addressed the same problem using a decentralized mixed linear programming model. Tang et al. [10] investigated this problem considering more practical situations when obstacles exist and mobile sinks can only move along certain paths in the field to collect data. The most recent work on network lifetime maximization using multiple cooperating mobile sinks is presented by Liang et al. [9]. He formulated the problem as optimizing h-hop-constrained multiple mobile sink movement and found the optimal trajectory for every sink using heuristics.

All of the abovementioned works formulates the data collection as an optimal path selection problem with different assumptions and constraints. A different method using intermediate mobile nodes is first proposed for WSNs by Shah et al. in [1]. They propose to use randomly moving "Data Mules" for data gathering. Data Mules in the sensor field are used as forwarding agents. The idea is to save energy by using single hop routing (from the sensor to the "Mule") instead of more expensive multi-hop routing (from the sensor to the sink). The "Mule" eventually approaches the sink and delivers all collected data to the sink. In this architecture, energy is traded off for latency. Wang et al. [8] investigates the performance of a large dense network with one MRN and shows that the improvement in network lifetime over an all-static network is upper bounded by a factor of four. Communication involving relay nodes in those approaches only considers sensor-to-relay and relay-to-sink scenarios. Relay-to-relay communication, which could be potentially useful, is neglected or underutilized. To the best of our knowledge, no work has considered the benefits of letting the data-collection-layer nodes communicate to form a network. One idea most similar to ours is presented in [3]. However, the relay nodes in [3] do not directly get data from sensors and they move to facilitate the communication between static data aggregation centers of different sensor clusters. We are dealing with a more dynamic situation where all data collectors are moving and we jointly consider load assignment, data collection (from sensors), and data routing (among MRNs) problems, in addition to the initial fully automatic network deployment issues.

3 Proposed Models

Before presenting our data collection scheme, we would first like to specify the general assumptions about the WSN model we use. We are considering a WSN with sensors that have limited energy and buffer. The MRNs are rechargeable mobile devices with much larger buffer space. And they are also supposed to have much longer communication range than sensors, while not to the extent of accessing backbone network directly. Finally, we assume the MRNs are equipped with GPSs.

Based on those assumptions, we propose to use MDCNet to collect and route data for sensors. The MDCNet is a middle-layer network between the sensor network and the sink (see Fig. 1). It's composed of MRNs that move independently in the field to collect and forward data for a certain number of sensors. Sensors access the MDCNet to deliver their data by contacting MRN in a single hop. The data uploaded to a MRN will be routed through the MDCNet towards the static sink. The MDCNet is generally a partially and intermittently connected mobile ad hoc network. Data uploaded to a MRN will be temporarily buffered at each MRN on its path to the sink. MRNs only communicate with their neighboring peers when they need to send data.

In order collect data effectively, the MDCNet has to satisfy several requirements. Firstly, the number of sensors that every MRN serves should be balanced to average their utilization rate and reduce sensor contention. Secondly, most of the sensors' data should be collected in time to avoid data loss caused by sensor buffer overflows. Thirdly, a reliable data relay protocol among MRNs should be developed to make sure that data uploaded from sensors would arrive at the sink safely. To satisfy the first requirements, we need to solve a load-balanced area partitioning problem, which constitute the first step of our data collection scheme. And we address the data collection and data transmission problems by developing corresponding communication protocols.

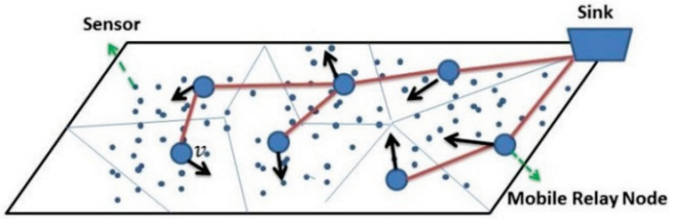


Fig. 1. Delay Tolerant Mobile Data Collection Network

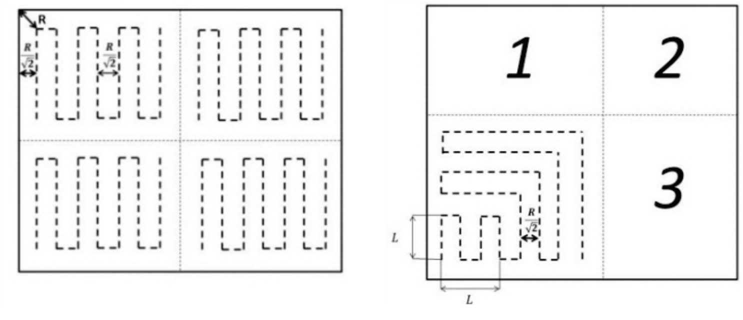
3.1 Load Balanced Area Partitioning

Assuming different degree of global knowledge, we propose two algorithms to solve the load-balanced area partitioning problem.

A. Deterministic Area Partitioning

The deterministic area partitioning (DAP) solution is based on the assumption that the distribution of the sensors is known in advance and that MRN's initial position can be deployed by a central administrator. Consider, for example, the case when the sensors are evenly distributed over a square area. Load balancing could easily be achieved by equally partition the region and assign one MRN to each of the partitions (as illustrated by Fig. 2(a)). In order for the sink in each square to provide full coverage of sensors in its assigned square, they are programmed to move in a snake-like manner as illustrated in Fig. 2. And the distance between parallel paths are set to be $R/\sqrt{2}$, where R is the communication radius of the sensor.

This simple solution to the area partitioning surely has some nice properties. It's easy to execute and would perfectly satisfy the load balance requirement. Latency is also controllable by varying the partition size. However, the assumption we made for the solution is not practical in most situations. The distribution of sensors is normally random and we generally do not know them in advance. Moreover, a centralized administration is not always feasible. Therefore, this scheme actually gives us an upper bound when all assumptions are valid or a lower bound otherwise.



(a) Deterministic Partition (b) Adaptive Search and Conquer

Fig. 2. Load Balanced Area Partitioning

B. Adaptive Search and Conquer

The *adaptive search and conquer* (ASC) strategy we present here assumes no prior knowledge of the sensor distribution and is totally distributed. We assume all the MRNs are located at the origin, say the bottom-left corner of the rectangular area at the beginning of network operation. MRNs incrementally enlarge their search space within a *target area* (TA) until they reach a predefined load factor (i.e. finding a certain number of sensors). Then they claim the area traversed so far as its *service area* (SA) and act as moving data collector for this SA. They will also notify other idle MRNs to search in the unexplored areas, if there are any. The search will not stop until the combination of all SAs cover the whole network area. After that, every MRN will move along certain paths within its SA to gather sensor data and relay them to the sink.

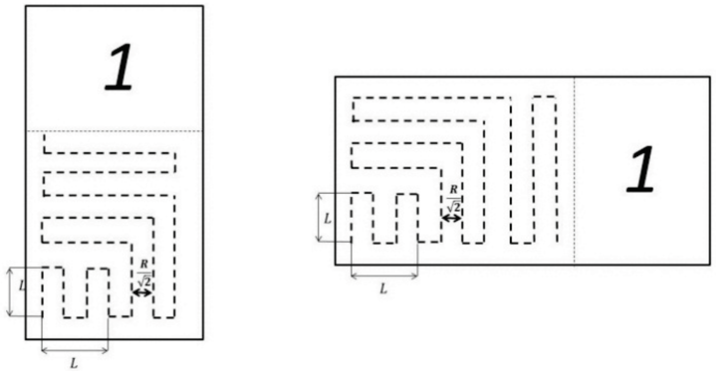


Fig. 3. Adaptive Area Partitioning Scenarios

In detail, every MRN would set a random timer in the beginning. The MRN whose timer expires first would start out from the origin and take the whole rectangle as its TA to search for sensors. The searching is done by broadcasting *DISCOVER* messages periodically and keeping a counter of the number of different sensors that reply with *ACK* messages. The MRNs do exhaustive search in the following manner (see Fig. 2(b) and Fig. 3) in order to provide full coverage. In the beginning, the moving path follows a snake-like pattern within an $L \times L$ square, where L is a preset parameter. If the MRN does not find enough sensors when it searched the initial square area, it begins to expand the searching square by $R/\sqrt{2}$ (recall that this step size is chosen to provide full coverage) every step and search along the border of the expanded square region. The reason for this searching strategy is that we want the conquered area for every MRN to approximate a compact square rather than a disproportioned long stripe. Once the MRN has found a certain number of sensors, it will claim the rectangular area it has traversed so far, and at the same time, it will broadcast a *NOTICE* message to notify idle MRNs of the location information of the unexplored areas in its TA. The idle MRNs who hear the *NOTICE* message would make a list (in case they hear multiple messages) of the unconquered areas and set a random timer for the first item of its unconquered area list. If it doesn't hear any other MRNs set out before its timer expires, it will broadcast a *TIMEOUT* message and set this area as its TA and search for sensors in the same way as the first MRN. Other idle MRNs will cancel the timer they set for the first item in their unconquered area list, remove it from the list and set a new timer for the current first item in the updated unconquered area list, if it's not empty. Based on squareness of TA, every time a MRN claims a SA, its TA will be divided into 2 or 4 parts (see Fig. 3 for 2-part scenarios), and this partition process is done recursively and distributedly through communication among MRNs until the whole sensor field has been covered.

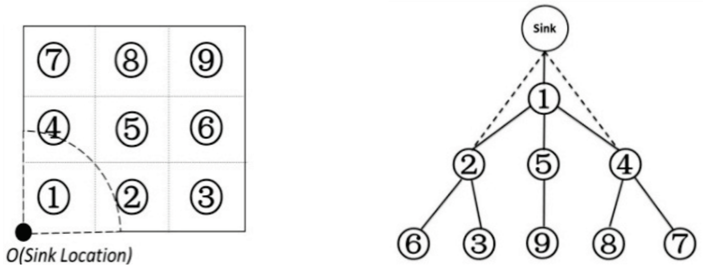
3.2 Data Collection Protocol

As soon as a MRN has conquered an area, it begins to move within its SA in a snake-like pattern (see Fig. 2(a)) to collect data from sensors. Specifically, the MRNs broadcast *HELLO* messages periodically. Sensor that hears the *HELLO* message will reply an *ACK* message when it needs to upload its data. After receiving *ACK* messages from the sensor, the MRN will look up the service history record to make sure that sensors do not get repetitive service within a short interval. If the sensor satisfies the service requirement, the MRN will stop to receive data from the sensor until all its data has been uploaded.

3.3 Data Relay Protocol

The MRNs forms a delay tolerant ad hoc network through which gathered data will finally get to the sink. For the two different area partitioning schemes described earlier, the data relay hierarchy is derived in different manners.

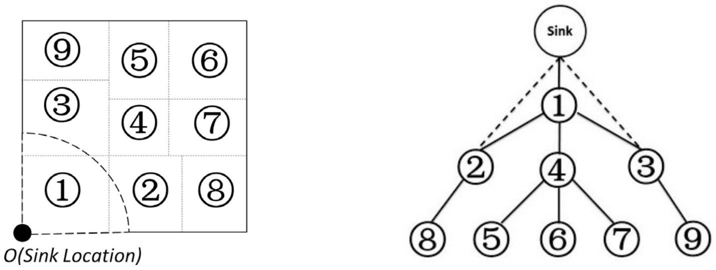
For DAP, since the partition is pre-determined, the level of a MRN in the relay hierarchy is also pre-derived based on the partition it is assigned to. In detail, the parent of a MRN is set to the MRN that is closest to the sink among its eight-neighborhood. For example, consider the partition showed in Fig. 4(a), the corresponding relay hierarchy is showed in Fig. 4(b). During network initialization, the administrator would set the parent information for each MRN.



(a) Deterministic partition example (b) Data relay hierarchy of partition (a)

Fig. 4. Deterministic Load-balanced Area Partition and Data Relay Hierarchy

While in the ASC strategy, the relay hierarchy is built automatically and dynamically when MRNs cooperatively communicate and search. The MRN that sets out first will set its parent to be the sink. Other MRNs will set their parent to be the MRN from whom they get their TA information. After the whole sensor field has been claimed, the hierarchical data flow structure will be completed. The data will flow in a bottom up fashion following the hierarchical structure. Consider the example shown in Fig. 5(a). MRN 1 is the first searching node. When it claims the left-bottom rectangle, it broadcasts a *NOTICE* message with the information of 3 unexplored areas. MRN 3, 4 and 2 won the competition of random timing, thus set their parent to MRN 1 and start their search in corresponding TAs. MRN 2, 3 and 4 further broadcast unexplored area information as they find enough sensors in their TA. In response to the new *NOTICE* messages, MRN 5 to 9 set out to search and finally claimed the remaining area. The corresponding data relay hierarchy is shown in Fig. 5(b).



(a) Adaptive partition example (b) Data relay hierarchy dynamically built for (a)

Fig. 5. Adaptive Load-balanced Area Partition and Data Relay Hierarchy

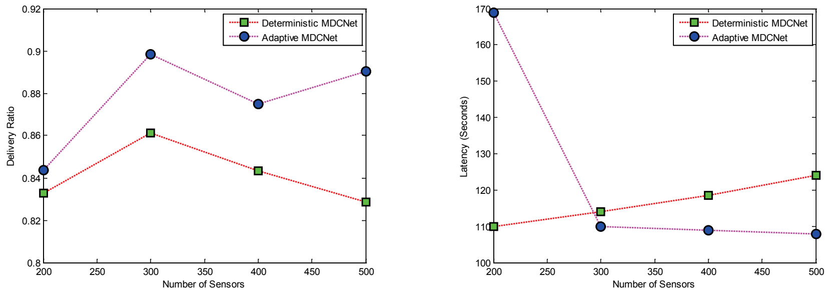
In addition to the parent address information, each MRN will also keep a flag indicating its closeness to the sink. Specifically, if the part of a MRN's SA is within the communication radius of the sink, the flag will be set to true (corresponding to the dotted line in Fig. 4(b) and Fig. 5(b)), which indicates that this MRN will deliver its data directly to the sink instead of sending to its parent. However, if this MRN in no means could directly communicate with the sink, it will send *HELP* messages periodically to its parent node to seek for help. Upon receiving the *HELP* message, the parent node will reply a *READY* message as a signal of its readiness to accept data. Both the child and the parent will then stop to complete the data transmission until they finish.

4 Experimental Evaluation

In this section, we will present the simulation results of our new data collection model. We implement the MDCNet in *ns-2.35* and carry out extensive experiments to evaluate the performance of our new model under various settings. Specifically, we set the size of the WSN to $100m \times 100m$.The communication range of the sensors and the MRNs are set to *7m* and *40m* respectively. The moving speed of the MRNs is fixed to $2m/s$. The initial energy of sensors is set to 100J and energy consumption for transmitting and receiving data are both set to $10^{-3}J/bit$. Sensor generates a *10-bit* packet every *0.1* second and temporarily stores it in its *10KB* local buffer.

We evaluate the performance of our data collection strategy with two important WSN QoS metrics: data delivery ratio and latency. The metrics are acquired when the first sensor depletes its energy (generally considered in the community as the WSN lifetime). In the first set of simulations, we vary the number of randomly-distributed sensors N from *200* to *500* and compare the performance of deterministic and adaptive scheme. The load factor parameter L_f for adaptive MDCNet is set to *40* and we use *4* MRNs for the deterministic scheme. The comparison of the two schemes is shown in Fig. 6.

We note that in both schemes, there is an increase of data delivery ratio when we change N from 200 to 300. This is because when N is below 300, the number of sensors each MRN serves has not reached full load. Further increase of sensors after full load point will demonstrate the better load balancing characteristics of the distributed MDCNet. In detail, for the centralized scheme, if we further increase the number of sensors, the delivery ratio will keep dropping as a result of overload. The overload effect also accounts for the steady increase in latency as shown in Fig 6(b). While in the adaptive scheme, since the load factor of each MRN is independent of the number of sensors, its actual load is insensitive to the increase of sensor density. And the number of MRNs is determined dynamically by the total workload. Thus, its data delivery ratio is higher than the centralized scheme in all cases. And the latency for the adaptive scheme decreases as sensor density gets higher, which is the result of a smaller SA. This result is consistent with our prediction that the deterministic approach is inferior when its assumptions are not valid.

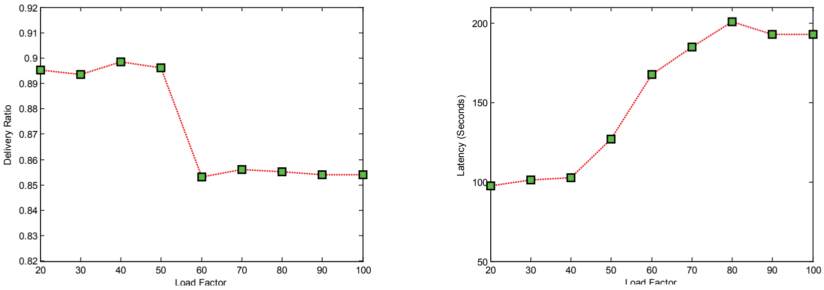


(a) Delivery ratio with varying N

(b) Latency with varying N

Fig. 6. Comparison between adaptive and deterministic MDCNet

The second set of simulations is designed to evaluate the effect of load factor on the performance of the distributed MDCNet. We use a uniform grid-pattern distribution of sensors to rule out the effect of randomness. In this simulation, we fix the number of sensors to 300 and vary L_f (Load factor) from 20 to 100. We evaluate the same metrics as we do in the first set of experiments. The results are shown in Fig. 7.



(a) Delivery ratio with varying L_f (b) Latency ratio with varying L_f

Fig. 7. Impact of load factor on delivery ratio, latency and number of MRNs

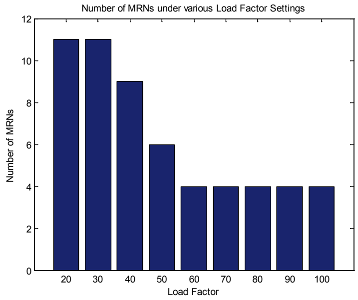


Fig. 8. Number of MRNs for different Load Factors

It can be noted that delivery ratio drops significantly when the load factor exceeds 50. And the latency also begins to increase drastically around 50. The behaviors of both metrics indicate that the maximum capacity of a MRN is around 50 sensors. Below the maximum capacity level, the latency and delivery ratio with respect to different load factor does not vary much. However, the close metrics are achieved at different costs (in terms of number of MRNs), which could be observed from Fig. 8. Considering all aspects, the optimal load factor should be between 40 and 50 and the choice should be a tradeoff between latency and cost.

5 Conclusion and Future Work

In this paper, we propose a new paradigm—the MDCNet for effective data collection in WSNs. The major contribution of our work is twofold. Firstly, the new data collection paradigm shows promising QoS for WSNs in various ways: it saves sensor energy by limiting sensor's communication to single hop and achieves good latency and delivery ratio by letting the MRNs form a dynamic ad hoc network. Thus the new concept of data collection network will open a new window for further research on improving QoS for WSNs. Second, we jointly consider load balanced sensor assignment, effective data collection and reliable data transmission problems and give an implementation of the new data collection paradigm in *ns-2* under different assumptions. On our basis, more advanced data collection solutions for the new paradigm can be motivated and developed under weaker assumptions. In particular, we are considering in our future work to further reduce assumptions such as the GPS for MRNs and the rectangularity of MRN's Service Area, and try to solve the MDCNet problem with more constraints (e.g. mobility of MRNs are constrained to certain paths due to the existence of obstacles).

References

1. Shah, R., Roy, S., Jain, S., Brunette, W.: Data MULEs: Modeling a Three-tier Architecture for Sparse Sensor Networks. In: Proc. of the 1st IEEE SNPA, 2003.
2. S. Gandham, M. Dawande, R. Prakash, and S. Venkatesan: Energy-efficient schemes for wireless sensor networks with multiple mobile base stations. In: Proceedings of IEEE GLOBECOM, Dec 2003.
3. Y.T. Hou, Y. Shi, H.D. Sherali and S.F. Midkiff: Prolonging Sensor Network Lifetime with Energy Provisioning and Relay Node Placement. In: SECON'05, pp. 295-304.
4. Jea, D., Somasundara, A., Srivastava, M.: Multiple Controlled Mobile Elements (Data Mules) for Data Collection in Sensor Networks. In: Proc. of the 1st IEEE/ACM DCOSS, 2005.
5. S. Basagni, A. Carosi, E. Melachrinoudis, C. Petrioli, and Z.M. Wang: Controlled Sink Mobility for Prolonging Wireless Sensor Networks Lifetime. In: Wireless Networks, vol. 14, no. 6, pp. 831-858, Feb. 2007.
6. Li, J. and Mohapatra,: Analytical modeling and mitigation techniques for the energy hole problem in sensor networks. In: Pervasive Mobile Computing 3, 3, 233-254, 2007.
7. Y. Shi and Y. T. Hou: Theoretical results on base station movement problem for sensor network. In: IEEE INFOCOM, 2008.
8. Wang, W., Srinivasan, V., Chua, K.C.: Extending the Lifetime of Wireless Sensor Networks through Mobile Relays. In IEEE/ACM Transactions on Networking, (2008), pp. 1108-1120
9. Liang Weifa, Luo Jun: Network lifetime maximization in sensor networks with multiple mobile sinks. In IEEE Conference on Local Computer Networks (LCN), 2011
10. S. Tang, J. Yuan, X. Li, Y. Liu, G. Chen, M. Gu, J. Zhao, and G. Dai: DAWN: Energy Efficient Data Aggregation in WSN with Mobile Sinks. In: IEEE International Workshop on Quality of Service, 2010.
11. 10. M. Di Francesco, S. Das, G. Anastasi: Data Collection in Wireless Sensor Networks with Mobile Elements: A Survey. In: ACM Transactions on Sensor Networks, 2011.
12. Fall K: A delay-tolerant network architecture for challenged internets. In: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM), pp.27–34, 2003.
13. Kien A. Hua and Fei Xie, "A Dynamic Stream Merging Technique for Video-on-Demand Services over Wireless Mesh Access Networks," in Proceedings of IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2010).