International Journal of Future Generation Communication and Networking
Vol. 9, No. 6 (2016), pp. 351-362
http://dx.doi.org/10.14257/ijfgcn.2016.9.6.33

Minimum Delay Query in Low-duty-cycle Sensor Networks

Zuzhi Fan
Department of Mathematics, Jinan University,
Guangzhou 510632, China
tfanzz@jnu.edu.cn

Abstract

Data query is a common communication pattern in sensor networks. However, due to the introduction of sleep latency, it is challenging to perform efficient query in low-duty-cycle sensor networks. In particular, the round-trip delay from the sink to source nodes and vice versa over the same forwarding path could be extremely long. In this work, westudy the delay optimization problem by jointly considering routing and sleep scheduling, then propose a holistic scheme to find the asymmetric routing paths for data query and response so that the round-trip delay could be minimized. We compared ourMinimum Delay Query (MDQ) deign with the state-of-the-art schemes. The evaluation results verify that our design can greatly reduce the query delay in low-duty-cycle sensor networks.

Keywords*: query delay; wireless sensor networks; low-duty-cycle*

1. Introduction

Low-duty-cycle operations have been widely adopted to reduce energy consumption of nodes in wireless sensor networks [1]. Under duty-cycling model, sensor nodes turn off their sense and radio periodically, saving large amount of energy consumed in idle listening state. However, the network connectivity becomes intermittent and time-dependent.To perform transmission, a sender has to wait the wake up of its receivers, leading to the introduction of sleep latency[2], which is extremely long in low-duty-cycle sensor networks.

Recently, research work have been proposed to reduce sleep latency in low-duty-cycle sensor networks. Most of them can be classified into two kinds according the pattern of data delivery. One is focused on the communication from the sink to source nodes in the field, such as data broadcasting [3-4], end-to-end communication [5-6]. The other is convergecast via either dynamic data forwarding [7-8] or time pipelining [9-10].Though these designs are proved to be effective under the given assumptions, none of these are concerned with data query, a two-way data delivery process.

In this work, we address data query problem in low-duty-cycle sensor networks, which is a bidirectional communication pattern. In a typical scenario, the sink issues a query and source nodes response with the required data.Due to sleep scheduling, the multihop communications in low-duty-cycle sensor networks could endure a comparable long period of waiting time.In particular, we observe that the sleep latency over the same forwarding path is not symmetric, indicating that data query cannot be regarded as the simple combination of two-way communication. Actually,the delay of query diffusion from the sink to the network device is quite different fromupward data collection.

Some existing work [12, 15] have exploited the delay optimization from the perspective of temporal control. Differing from them, we explore the utilization of spatial diversity to reduce query delay in low-duty-cycle networks. The intellectual contributions in this work include: (i) we analyze the round-trip delay and demonstrate that the two way sleep latencies are mutual complementary. In other word, the reverse data forwarding

ISSN: 2233-7857 IJFGCN

along the delay-minimized path usually has the maximum latency; (ii) we devise a scheme to build asymmetric data forwarding paths between source nodes and the sink. Thus, the query is forwarded along the delay-minimized path while data packets are returnedalong a different delay-optimized path; (iii) the extensive simulations are conducted and the results validate that our design can greatly reduce round-trip delay and achieve suboptimal transmissions compared with the baseline algorithms.

The remainder of the paper is organized as follows. Section 2 introduces some related work. In Section 3, we give the network model and the problem definition. In Section 4, we describe our main design by introducinga routingmetric and concrete implementation. The proposed scheme are evaluated and compared with other schemes in Section 5 and Section 6 concludes this paper.

2. Related Work

Low-duty-cycle model has been shown an energy-efficient method for cyber-physical applications which require long and sustainable existence. At the cost of reduced energy consumption, the sleep latency is the paramount objective in low-duty-cycle sensor networks.

Recently, a few work have been proposed for the purpose of delay optimization in low-duty-cycle networks [3-10]. Broadcasting or flooding is a kind of one-to-many communication. In [3], Xu *et al*. address the schedule of nodes wake-up in order to reduce the number of transmissions. Assuming that wireless link is unreliable, Guo *et al*. [4] study the opportunistic flooding in low-duty-cycle sensor networks. In particular, a DAT-based flooding tree is built so that the sender can make probabilistic forwarding decisions based on the schedule of next hop. In [16], Zhu *et al*. take link correlation into consideration and propose correlated flooding for low-duty-cycle sensor networks, in which the collective acknowledgement is implemented so that the number of transmissions can be greatly reduced. Differing from all these design, Gu *et al*. [5] present the approach to bind the sink-to-nodes delay in duty-cycling sensor networks. In specific, the working schedules of nodes are dynamically augmented or decreased based on the harvested energy. Besides temporal control, Gu *et al*.[6] also discuss spatial control method by introducing more sink nodes into network, decreasing the average length of forwarding path and communication delay between source nodes and corresponding sink.

All aforementioned work are based on the assumption that the duty cycle of sensor nodes could be scheduled at will. DSF [7] is the first design to handle unreliability of wireless link and sleep latency based on the space diversity. In DSF, each node selects a set of neighboring nodes instead of one as potential forwarders. Once failed, the sender can choose next candidate to retransmit rather than wait for a whole working period. The selection of forwarding set could be made forachieving various objectives, such as delivery ratio, delay or energy efficiency.Based on the asynchronous duty-cycling mode, Landsiedel *et al*. [8] propose opportunistic routing for duty-cycling sensor network, in which each node selects the first wake-up node as next-hop,decreasing the waiting time. Cao *et al*.[9] combine the streamed pipeline with dynamic switching method in order to handle the unreliable transmission for source-to-sink data delivery.

In sensor network, query is one of significant data acquisitiveapproaches, involving two-way communication [11]. In [13], Guo *et al*. consider a two way communication pattern in event-monitoring applications, where the alarm message indicating critical event is firstly delivered to the center node and then broadcasted to the whole network. In the implementation, two paths areseparatelybuiltwith corresponding sleep scheduling policies for uplink transmission and downlink broadcasting. Silva *et al*. [14] study the spatial query for sensor network with duty-cycle operations, emphasizing on the in-network processing. Accordingly, a location-based routing mechanism is presented for the data query and response.

To the best of our knowledge, circular pipelining, CP [15] is the first work focused on the delay optimization in low-duty-cycle sensor network. They prove that the minimal round-trip delay in simple circular path equals to the working period, while other non-circular path could be decomposed to multiple simple circular paths. As a result, the sleep scheduling problem can be transformed to finding the disjoint paths between the sink and source nodes. However, itassumes that the duty-cycle of sensor nodes is predefined and fixed in the runtime.Both DESS [12] and CP [15] are designed with time pipelining, which is totally different from our design, aiming at the delay optimization from the view of spatial diversity.

3. Preliminary

In this work, we assume a sensor network consisting of one sink and large amount of nodes working in low-duty-cycle model. We first present network model with duty-cycling operation and related assumptions, then formulate the problem.

3.1. Network Model

Under duty-cycling mode, a sensor node is either in an active or a dormant state at any point of time. In active state, the node can sense, transmit and receive data packets; while in dormant state, itturns off all function units except a timer to wake itself up. Usually, each node has its own individual working schedule to ensure the network coverage and connectivity. In other word, nodes may wake up asynchronously so that the network connectivity is intermittent. Thus, a node has to wait its receiver to wake up if it wants to transmit a packet.

In the real deployment, the working schedules of nodes are usually periodic [18]. Let the common duration of a period be T, it can be divided into a series of time slots with the same length (τ). The working schedule of a node *i* can be represented as the set of active slots, *i.e.*,$W_i = \{t_1, t_2, \ldots, t_n\}$, where t_i is the active time slot and n is the number of active slot within T. For example, the working schedule of node in Figure 1 is$W_i = \{2,5\}$.

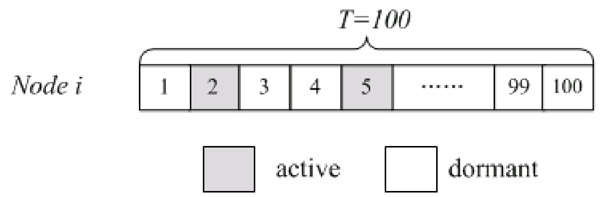

Figure 1. The Working Schedule

In always-on network, the delivery delay comprises of process delay, transmission delay and propagation delay, normally in the order of milliseconds. However, the delay in duty-cyclingsensor network is dominated by sleep latency, which is normally proportional to the number of waiting time slots. Noticeably, the value of a single time slot, τ is 20~40ms, which is much longer than traditional delay.

3.2. Assumptions

In this work, we make the following assumptions:

(1) The network is synchronized so that each node knows the working schedules of its neighbors and when it can send a packet.

(2) The transmission of a packet and its acknowledgement can be accomplished in one time slot. For example, suppose the length of a time slot is 20ms, a MicaZ

node with CC2420 radio chip can transmit a packet with the size 64 bytes about 10 times within a time slot.

(3) As aforementioned, the sleep latency is the dominating factor in low-duty-cycle sensor network. The other delay, such as process, transmission, propagation is trivial in the literature. Consequently, the sleep latency and delivery delay, latency can be used interchangeably.

3.3. Problem Description

For clarify, suppose that each node wakes up only once during a whole period of time T. Later, we will show how to extend our design to support multiple wake-ups. Given the wake-up slots of one sender i and its receiver j are t_i,t_j, respectively, the one-hop sleep latency from i to j can be calculated as:

$$\mathrm{d_{ij}} = \begin{cases} t_j - t_i & t_j \geq t_i \\ (t_j - t_i) + T & otherwise \end{cases} \quad (1)$$

We observe that two-way delay is not symmetric along with the data forwarding path. Taking one-hop delivery in Figure 2 as an example, let T be 100, the sender i wakes up at 2 and the receiver j wakes up at 4, i.e., $W_i = \{2\}, W_j = \{4\}$. Thus, the one-hop delivery delay from node i to node j is $d_{ij} = 4 - 2 = 2$. However, if node jsendsdata to node i, the corresponding delay is $d_{ji} = (2 - 4) + T = 98$. That is, node j has to wait for a whole period of time (T) till node i wakes up in next round. In fact, we have the following observation about two-way delay,$d_{ij} + d_{ji} = T$. That is, the round-trip delay is mutually complementary in one-hop data delivery.

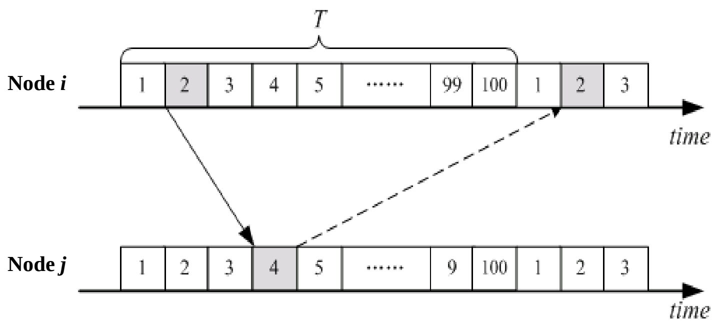

Figure 2. One-hop Delivery Delay in Low-Duty-Cycle Sensor Networks

Now, we consider the query process in multihop forwarding path. At first, we give the following definition:

Round-trip Delay: the round-trip delay,D_i^j is the period of waiting time from the moment that node i sends out a query till it receives the response from node j.

Generally, the round-trip delay is comprised of delays over two directional transmissions. Let the forwarding path of data query and response be P_1, P_2, we have:

$$D_i^j = D_{ij}(P_1) + D_{ji}(P_2). \quad (2)$$

Figure 3 illustrates a linear topology with streamlined working schedule, which can achieve the minimal delivery delay from the sink to node D according to design in [12]. Here, route path from sink to node D, P_1 is {Sink, A, B, C, D} and the delivery delay is 4. However, if the response data is delivered over the reverse route,P_2= {D,C,B,A, Sink}, the trip delay is 4T, which is much longer compared with one-way delivery delay.

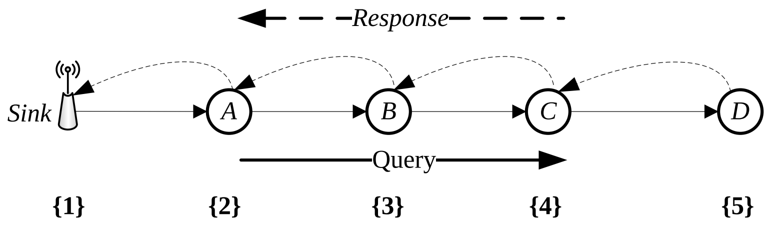

Figure 3. Data Query over Multiple-Hop Path

In this work, we study the data query problem to minimize round-trip delay in low-duty-cycle sensor networks. As a result, the sink can send query to source nodes and acquire the returned data as fast as possible. The key challenge is how to build appropriate forwarding paths for all sensor nodes.

4. Minimum Delay Query Scheme

In this section, we propose a minimum delay query (MDQ) scheme for low-duty-cycle sensor networks.

4.1. Delay Model

As we discussed, the sleep latency is the most important element for low-duty-cycle sensor networks. Therefore, it is vital to evaluate the minimal query delay and response delay for given multihop forwarding path. Specially, we represent the minimal query and response delay between the sink and node i as $\mathrm{Q}_i^{min}, \mathrm{R}_i^{min}$, respectively.

At first step, for those nodes that are one-hop away from the sink, the initial query delays are equal to the sleep latencies between them. Recursively, the query and response delayof descending nodes (farther nodes) can be calculated according to dynamical programming. Given the minimal query delay of one-hop neighbor j, the minimal query delay of nodei can be calculated by

$$\mathrm{Q}_i^{min} = \min\{\mathrm{Q}_i^{min}, Q_j^{min} + d_{ij}\}. \tag{3}$$

Similarly, the response delay could be computed:

$$\mathrm{R}_i^{min} = \min\{\mathrm{R}_i^{min}, R_j^{min} + d_{ji}\}. \tag{4}$$

To further explain the above process, we illustrate the computation with a simple example shown in Figure 4. At first, we can obtain the delay of one-hop neighborsfor sender s(Figure 4a):

$$\mathrm{Q}_a^{min} = 2,\ \mathrm{Q}_b^{min} = 7.$$

According to the equation (3), we have the following result:

$$\mathrm{Q}_c^{min} = \min\{\mathrm{Q}_a^{min} + d_{ac},\ \mathrm{Q}_b^{min} + d_{bc}\} = 5.$$

Similarly, we can calculate the response delays for node a,b and c (Figure 4b),

$$\mathrm{R}_a^{min} = 8,\ \mathrm{R}_b^{min} = 3, \mathrm{R}_b^{min} = 5\ .$$

Based on the results, each node can determine the optimal paths for both query and data response. As shown in Figure 4, the optimal query path from the sink s to node c is {s, a, c}, while the corresponding response path is {c, b, s} (both are denoted as thicker lines).

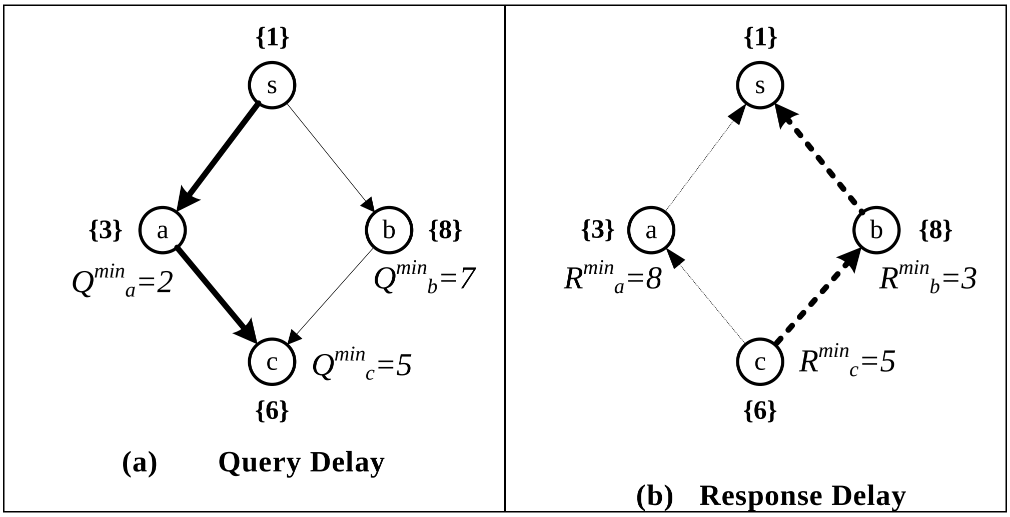

Figure 4. Calculation of Query and Response Delay (T=10)

4.2. Design Overview

In the above calculation, we have the following observation: for given source and destination node, the one-way sleep latency depends on the number of inverted pair of nodes, instead of wake-up slots along the forwarding path.To present, we give the definition of ***i**nverted **n**ode **p**air*formally.

Definition*INP*: For a pair of neighboring node, if node i intends to send packet to node j and their wake-up slots satisfies $t_i>t_j$, they called ***i**nverted **n**ode **p**air*. In the following, INP is used as a metric to represent the number of inverted node pairs along a forwarding path.

Based on the observation, we have the following lemma.

Lemma 1. For given source node s and destination t, let $P_1 = \{s, a_1, a_2, \dots, a_m, d\}$ and $P_2 = \{s, b_1, b_2, \dots, b_n, d\}$ be two disjointed paths, the one with smaller **INP**has the smaller delivery delay.

Proof: In order to validate the observation, we assume a topology shown in Figure 5. Suppose the time slots along P_1,$P_2$are $\{t_0, t_1, t_2, \dots, t_m, t'\}$ and $\{t_0, t'_1, t'_2, \dots, t'_N, t'\}$ respectively, we have the following results according to equation 1.

$$\mathrm{D}_{sd}(P_1) = \sum_{i=0,j=i+1}^{m} d_{ij} = (t_1 - t_0) + (t_2 - t_1) + \cdots + (t' - t_m) + MT$$
$$=(t'\text{-}t_0)\text{+MT}. \quad (5)$$

Here,M($\mathrm{M} \le \mathrm{m} + 1$) is the number of inverted wake-up pairs. Similarly, the delay through P_2 is $\mathrm{D}_{sd}(P_2) = (t' - t_0) + NT$, N($\mathrm{N} \le \mathrm{n} + 1$).Obviously, the delay between s and d depends on only the values of INP. Consequently, we can select the path with smaller INP. For node-jointed paths, we can prove by dividing it into several sub-paths.

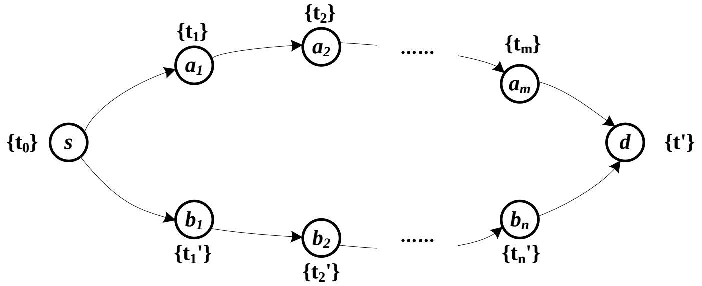

Figure 5. Example of Lemma 1

Lemma 2. For given source node s and destination d, the minimal round-trip delay depends on INP along the circular path, instead of the concrete wake-up slots.
Proof: According to equation 2, the round-trip delay between node s and node d is $D_s^d = D_{sd}(P_1) + D_{ds}(P_2)$.Based on the conclusion in lemma 1, we have:

$$D_s^d = D_{sd}(P_1) + D_{ds}(P_2) = (\mathrm{t}' - \mathrm{t}_0) + \mathrm{M}'\mathrm{T} + (\mathrm{t}_0 - \mathrm{t}') + \mathrm{N}'\mathrm{T} = (\mathrm{M}' + \mathrm{N}')\mathrm{T}, \quad (6)$$

which finishes our proof.

Consequently, we can use INP instead of sleep latency as a metric to build query path from the sink to source nodes. As shown Figure 4(a), we can calculate the INP of left path, INP({s, a, c})=0, which is smaller than right path, INP({s, b, c})=1. In Figure 4(b), the right path (with smaller INP) is the optimal one for response delay.

4.3 Minimum Delay Query Algorithm

Intuitively, we can build a minimum-INP spanning tree rooted at the sink, which can be accomplished with traditional single-source shortest path algorithm, i.e., Bellman-Ford algorithm. In the case, the query could be delivered to all nodes as fast as possible. For the returned data packets from source nodes, one may argue that the reverse spanning trees rooted at the source nodes could be established similarly. It is noticeable that such a naïve design is not scalable in large-scale sensor networks, incurring too much overhead and energy consumptions.

In this work, we propose a holistic scheme to find the reversed response path for each node while building minimum-INP spanning tree. The pseudo code of our algorithm is presented in Algorithm 1. In the implementation, the sink broadcasts a QUERY_BUILD beacon including hop count, INP and RINP (reversed INP from source node to the sink), all initialized as 0. On receiving the beacon from its parent, the node will update its own INP and RINP to achieve the minimal INP and RINP. Noting that, given wake-up slots of sender and receiver, both INP and RINP can be calculated (See Line 7-8). Thus, the minimized INP and RINP are determined recursively. When the current node makes its own forwarding decision, it would broadcast the beacon with updated values. Iteratively, the next-hop of query and response could be selected. The converging time of the algorithm is O(HD), where H is the hop counts from the sink to current node and D is the maximum number of neighboring nodes.

Algorithm 1.Pseudocode of Asymmetric Routing at Node i.

1. **Input: the neighboring set of node i, N_i;**
2. **Input: the wake-up slot and INP, RINP of neighbor j, t_j;**
3. **Input: the hop count of neighbor j, h_j;**
4. **Output: the next-hop for queryor response, N_{up}, N_{down};**
5. On Received QUERY_BUILD(t_j, h_j, INP_j, $RINP_j$):

If(h_j +1>h_i) discard beacon;
If(t_j>t_i) tempINP = INP_j+1;
    Else tempRINP = $RINP_j$+1;
$INP_i \leftarrow \min(INP_i, tempINP)$;
$RINP_i \leftarrow \min(RINP_i,, tempRINP)$;
h_i = h_i+1;
Update N_{up} if a smaller INP is obtained;
Update N_{down} if a smaller INP is obtained;
Broadcast QUERY_BUILD with purple (t_i, h_i, INP_i, $RINP_i$):

Figure 6 demonstrates the spanning tree with minimum INP and the corresponding RINP. Actually, there may be multiple concurrent paths which have the same INR values. For example, the data delivery from node F to sink has two paths, {F, E, D, G, Sink} and {F, H, G, Sink}, which have the same INR (1). In the case, we can take hop count as reference to remove the redundant paths. In specific, the path with smaller hop count would be selected, leading to smaller transmissions.

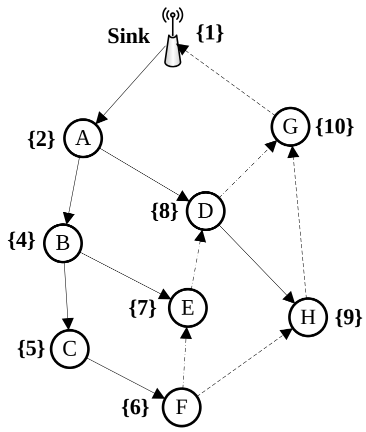

Figure 6. Minimum-delay Spanning Tree and Response DAG

It is necessary to address some other issues related to our design. At first, we assume that the wireless links are perfect in above sections, which is not always true in the real scenarios. In the case, we can use the metric presented in [17] to calculate the expected delay with unreliable link model. On the other hand, it is easy to extend our design to support multiple wake-ups during one period. When a node receives the broadcasted beacons from its neighbors, it will calculate the delay for each wake-up slot and select the minimized INP from the obtained results.

5. Performance Evaluation

In this section, we verify our design with large-scale simulations developed in C++. As far as we know, none of design is focused on the data query from the view of spatial diversity. Thus, we compare our design, MDQ with two modifiedrouting schemes.

- Shortest Path Query (termed SPQ): This is the improved version of traditional shortest path routing, with which both the query and data are delivered along with the shortest path.
- Minimum Latency Query (termed MLQ)[18]: The original version is proposed to find the minimum latency route. In this work, we revised it to conduct two-way query. In specific, each node selects forwarding path with minimum sleep latency and the response data is forwarded via the same path.

In the simulation, we randomly deploy 200 sensor nodes in a square field with size from 50mx50m to 200mx200m. The transmission range is set to 15m. The neighboring node can communicate with each other only when they are both awake. The sink sends query to all nodes in the network. We evaluate the average round-trip delay and the corresponding transmission counts. The following results are based on the statistical results with 10 randomly-generated network topologies with various seeds.

5.1. Round-trip Delay and Transmission Counts

Figure 7 shows that the CDF (cumulative distribution function) of round-trip delays for three algorithms. It can be observed that more than 99% of data queries are finished in less than 300 time slots for our design. The two other designs have delays as much as 3 times longer than MDQ. For example, the delays for SPQ and MLQ are 900 time slots in order to achieve 99% data delivery. BothSPQand MLQ are designed in order to minimize the one-way delay. As we discussed, MDQ is based on the delay optimization of two-way data delivery. Another observation is that round-trip delay is always a multiple of T, regardless of the paths and the algorithms, which are consistent with our proof (see equation 6).

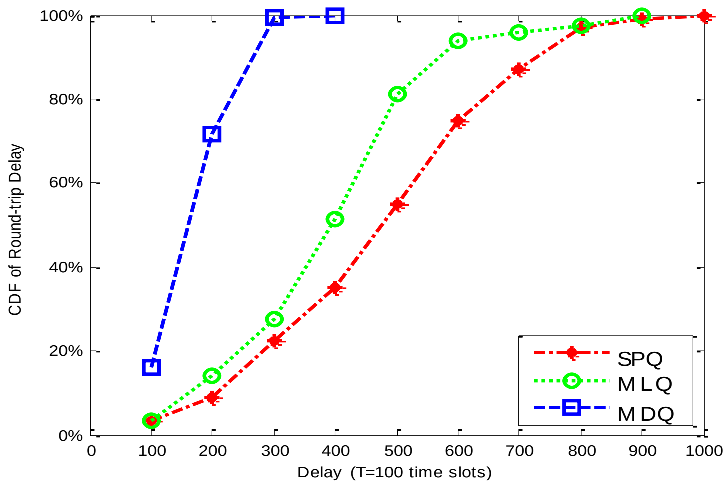

Figure 7. The CDF of Round-trip Delays

Figure 8 shows that the number of round-trip transmissions is comparable for three algorithms. With shorter forwarding path, both MDQ and MLQ have smaller transmission counts than SPQ, saving more consumed energy.

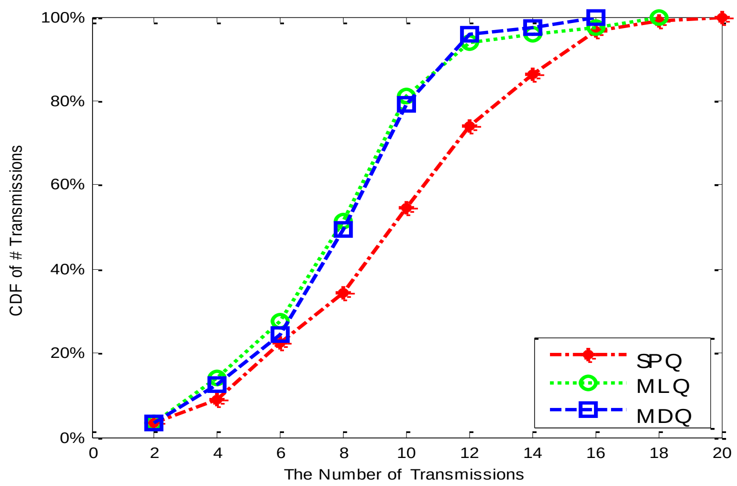

Figure 8. The CDF of Number of Transmissions

5.2. The Impact of Network Size

In this section, we validate our design under varied network size. Figure 9 shows that delays of three schemes increase with the network size. For example, the delay is 3.5 times for both SPQ and MLQ while the length of network size increases from 50 to 200. Meanwhile, there is anapparently increasing (about 6.5 times) for MDQ. The reason behind is that it is hard even impossible to find an appropriate circular path in sparse network. In fact, we observe that some nodes are disconnected from the network, leading to transmission failure.

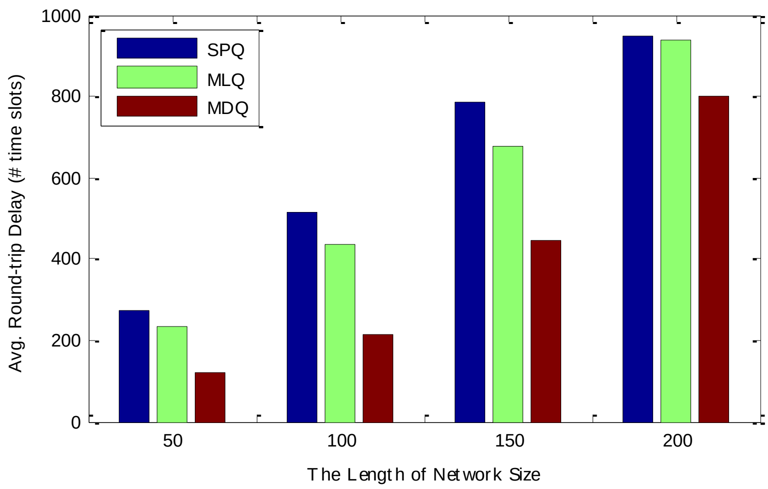

Figure 9. Delays vs. Network Size

Figure 10 shows that the transmission counts increase proportionally with the network size. However, SPQ has more transmissions compared with MLQ and MDQ, because the data query withSPQ usually has a longer forwarding path (in hop-count).

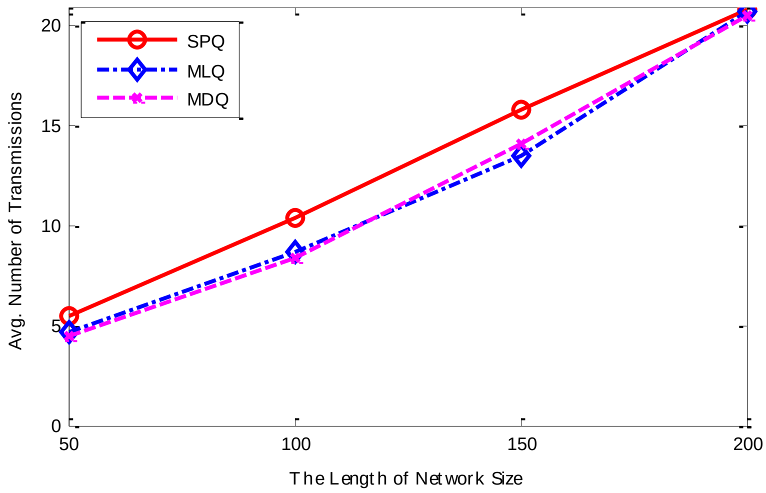

Figure 10.The Number of Transmissions vs. Network Size

5.3. The Impact of Working Period

In this section, we testify how the working period has impacts on the delay. As shown in Figure 11, the round-trip delays increases linearly with the working period for SPQ, MLQ and MDQ. For example, the average delay of SPQ increases from 261 to 1248 while the working period changes from 50 to 250. Nevertheless, the round-trip delays of MDQ are much less than other two schemes. That is, the round-trip delay depends on the number of inverted node pairs. We don't compare the performance of transmissions count, which is independentof working period.

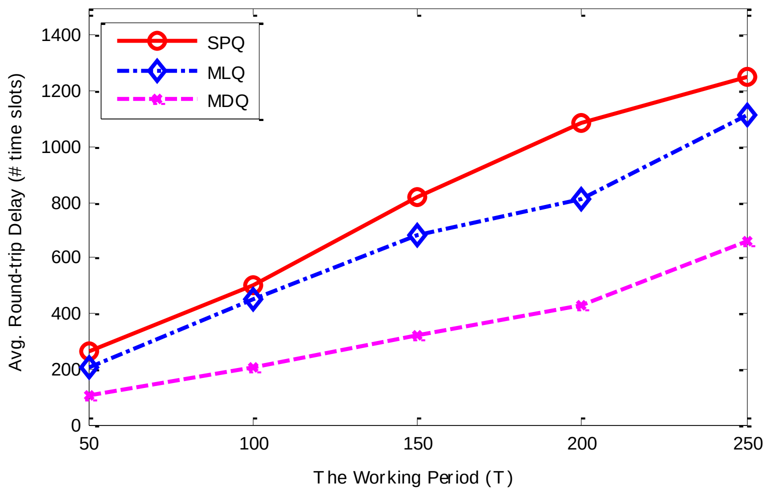

Figure 11. Delays vs. the Working Period (T)

6. Conclusion

In this work, we study the efficient data query problem in low-duty-cycle sensor networks, which is quite different from existing one-way data delivery. Based on our observation, the delays over one-hop communication are mutual complementary. Our analysis shows that the round-trip delay is only related to the number of inverted node pairs in the forwarding path. We proposed and implemented Minimum Delay Query (MDQ) for low-duty-cycle sensor networks. The performance evaluation shows that MDQ can greatly reduce round-trip delay but without incurring too much transmissions.

Acknowledgments

This work was supported in part by NSFC under grant No. 61373125, 61572233, Natural Science Foundation of Guangdong Province of China under grant No. 2016A030313104 and the Fundamental Research Funds for the Central Universities under grant No. 21615442.

References

[1] W. Ye, J. Heidemann and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks", Proceedings of IEEE INFOCOM, vol. 3, **(2002)**, pp. 1567- 1576.

[2] S. He, J. Chen and Y. Sun, "Coverage and Connectivity in Duty-cycling Wireless Sensor Network for Event Monitoring", IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 3, **(2012)**, pp. 475-482.

[3] L. Xu, J. Cao, S. Lin, H. Dai, X. Wu and G. Chen, "Energy-efficient broadcast scheduling with minimum latency for low-duty-cycle wireless sensor networks", Mobile Ad-Hoc and Sensor Systems (MASS), 2013 IEEE 10th International Conference, **(2013)**, pp. 163-167.

[4] S. Guo, L. He, Y. Gu, B. Jiang and T. He, "Opportunistic Flooding in Low-Duty-Cycle Wireless Sensor Networks with Unreliable Links", IEEE Transactions on Computers, vol. 63, no. 11, **(2014)**, pp. 2787-2802.

[5] Y. Gu and T. He, "Bounding Communication Delay in Energy Harvesting Sensor Networks", Proceedings of ICDCS, **(2010)**.

[6] Y. Gu, T. He, M. Lin and J. Xu, "Spatiotemporal Delay Control for Low-Duty-Cycle Sensor Networks", Proceedings of IEEE RTSS, **(2009)**.

[7] Y. Gu and T. He, "Dynamic switching-based data forwarding for low-duty-cycle wireless sensor networks", IEEE Transactions on Mobile Computing, vol. 10, no. 12, **(2011)**, pp. 1741-1754.

[8] O. Landsiedel, E. Ghadimi, S. Duquennoy and M. Johansson, "|Low power, low delay: opportunistic routing meets duty cycling", Proceedings of IPSN, **(2012)**, pp. 185-196.

[9] Y. Cao, S. Guo and T. He, "Robust Multi-Pipeline Scheduling in Low-Duty-Cycle Wireless Sensor Networks, Proceedings of IEEE INFOCOM, **(2012)**.

[10] X. Wang, X. Wang, G. Xing and Y. Yao, "Dynamic duty cycle control for end-to-end delay guarantees in wireless sensor networks", 18th International Workshop on Quality of Service (IWQoS), **(2010)**, pp. 1-9.

[11] G. Johannes and S. Madden, "Query processing in sensor networks", IEEE Pervasive Computing 1, (**2004)**, pp. 46-55.

[12] G. Lu, N. Sadagopan, B. Krishnamachari and A. Goel, "Delay efficient sleep scheduling in wireless sensor networks", Proceedings of IEEE INFOCOM, **(2005)**.

[13] P. Guo, T. Jiang, Q. Zhang and K. Zhang, "Sleep scheduling for critical event monitoring in wireless sensor networks", IEEE Transactions onParallel and Distributed Systems, vol. 23, no. 2, **(2012)**, pp. 345-352.

[14] D. S. R. Ilídio, D. F. Macedo and J. M. S. Nogueira, "Duty cycle aware spatial query processing in wireless sensor networks", Computer Communications, vol. 36, no. 2, **(2013)**, pp. 149-161.

[15] J. Li, S. M. Kim and T. He, "Circular Pipelining: Minimizing Round-Trip Delay in Low-Duty-Cycle Wireless Networks", The 22th IEEE International Conference on Network Protocols (ICNP '14), **(2014)**.

[16] T. Zhu, Z. Zhong, T. He, and Z. L. Zhang, "Achieving Efficient Flooding by Utilizing Link Correlation in Wireless Sensor Networks", IEEE/ACM Transactions on Networking, vol. 21, no. 1, **(2013)**, pp. 121-134.

[17] Z. Fan, "Delay-Driven Routing for Low-Duty-Cycle Sensor Networks", International Journal of Distributed Sensor Networks, Article198283, **(2013)**, 11 pages.

[18] L. Su, C. Liu, H. Song and G. Cao, "Routing in intermittently connected sensor networks", In Network Protocols, ICNP, IEEE International Conference, **(2008)**, pp. 278-287.

