EURASIP Journal on
Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

# Clustering-based routing for top-k querying in wireless sensor networks

Shangfeng Mo[1,2,3], Hong Chen[1,2*] and Yinglong Li[1,2,3]

**Abstract**

A large-scale wireless sensor networks (WSNs) can be deployed for top-k querying applications. There are many top-k querying algorithms which are based on traditional routing approaches. In this article, we proposed a clustering-based routing for top-k querying (CRTQ) in WSNs to save the energy consumption and extend the network lifetime. The proposed scheme consists of two parts: one is the cluster formation algorithm; another part is the inter-cluster choosing the relay cluster head algorithm. Moreover, we adopt a corresponding dynamic clustering algorithm. Our experimental result shows that CRTQ substantially outperforms the existing tree-based approaches and clustering-based approaches in terms of both energy consumption and network lifetime.

**Keywords:** WSNs, clustering-based, routing, top-k querying.

## 1. Introduction

Wireless sensor networks (WSNs) are a combination of sensing technology, embedded computing technology, distributed information processing and communication technology [1,2]. All sensor nodes can collaborate with each other to monitor, sense, and collect the information of environment or monitoring object, and to send the needed information to user. The technology of WSNs is considered as one of the most important technologies in the twenty-first century, and it will have a far-reaching impact on the human life in the future.

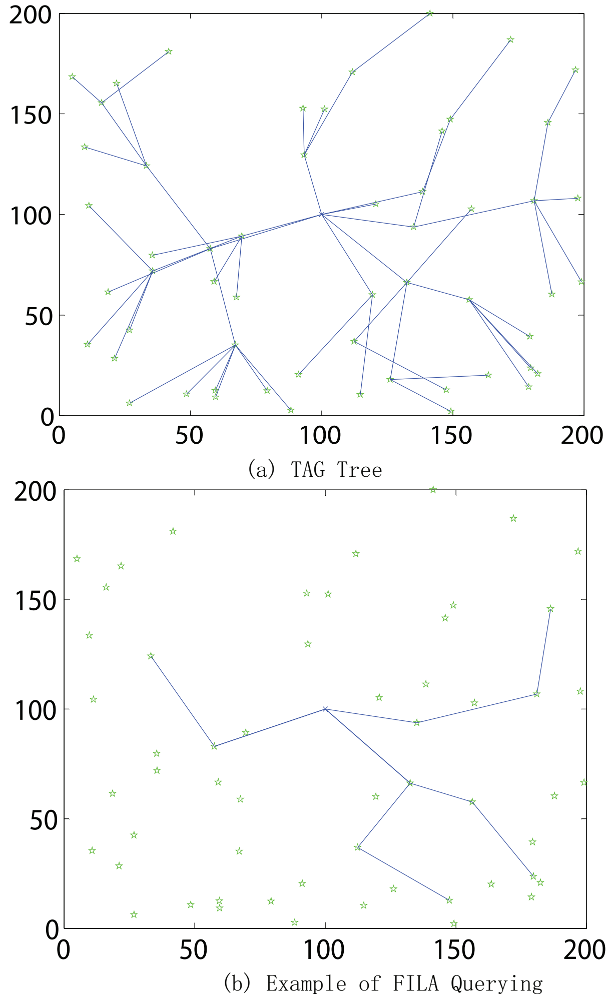
From the logical view of the data, WSNs can be viewed as a distributed database. The data management system of WSNs manages perceptible data from the monitored area and answers queries from users or applications.

Continuous top-k querying is defined as the algorithm in which the sink continuously requests the list of $k$ sensor nodes with the highest (or lowest) readings at every sampling period. The characteristics of a top-k querying are to reduce the amount of data transmission and thus save energy consumption. If only parts of nodes forward their readings to the sink, then it can also obtain the top-k result correctly, which can prolong the network lifetime and be used in most of top-k querying algorithms.

In recent years, there are many top-k querying algorithms (e.g., FILA [3], POT [4], PRIM [5], PRIM-c [6]), which consider how to reduce the amount of transmission data, thus reducing energy consumption and prolonging network lifetime. These algorithms are based on traditional routing algorithms with little change to the routing algorithms. For example, FILA and PRIM are based on the TAG [7] routing tree; PRIM-c is based on the hierarchical clustering and routing algorithms, such as LEACH [8], and HEED [9]. As shown in Figure 1a, the sink (the base station) node is located at the center, and ordinary nodes self-organize into a tree structure which is called TAG routing tree. Figure 1b shows an example of FILA querying, in which some nodes forward their readings to the sink (the base station) along the TAG routing tree. The top-k querying based on TAG has the following shortages: first, the depth of TAG is limited, and it is not suitable for large-scale WSNs; second, the node near the sink will take more relay task than other nodes, as well as consume more energy. Hence the nodes near the sink will exhaust their energy faster and destroy the connectivity of network.

We consider designing a new routing algorithm according to the characteristics of a top-k querying, and combine the current excellent top-k querying algorithms based on our new routing algorithm, which maybe outperform the top-k querying algorithms based on

* Correspondence: chong@ruc.edu.cn
Full list of author information is available at the end of the article



(a) TAG Tree

(b) Example of FILA Querying

**Figure 1 TAG tree and FILA querying**. (a) TAG Tree; (b) Example of FILA Querying.

traditional routings. We designed a new clustering and routing algorithms for top-k querying, which is named Clustering-based Routing for Top-k Querying (CRTQ). The CRTQ is defined as the order-sensitive exact top-k set querying of FILA based on our clustering routing algorithm. Our experimental results show that CRTQ outperforms FILA, and also outperforms the algorithm which combined other cluster-based routing protocol with FILA too.

The contributions of CRTQ are as follows. We introduce a competition-based clustering algorithm. Cluster heads with less residual energy than the average residual energy of all nodes in the network have smaller cluster sizes, and thus they can preserve more energy for inter-cluster data forwarding. In cluster formation phase, we obtain some information about the neighbor cluster heads, which will be used to find a neighbor cluster head which is closer to the sink (the base station) as a relay node, and avoid excessive use of probe-based algorithm. In our scheme, the clustering time is dynamic, which will balance the energy consumption among sensor nodes in the WSNs. CRTQ does not need to be predeployed, which simplifies system deployment. Finally, we also achieve an obvious improvement on the network lifetime.

The remainder of this article is organized as follows. Section 2 summarizes related studies. Section 3 introduces basic notations and assumptions. Section 4 describes the proposed CRTQ scheme in detail. Section 5 presents experimental results. Section 6 concludes the article.

## 2. Related work

We studied many top-k querying algorithms and routing protocols for WSNs, which have their own core mechanisms, characteristics, advantages, and disadvantages.

First, we will show some top-k querying algorithms.

FILA [3] is a suitable algorithm for continuous querying. At every sampling point, if the newly sensed data of a node does not change beyond the filtering window, the data will not be sent to the sink. If the sensed data come into the filtering window of other nodes, the sink will broadcast to all nodes in the WSNs and acquire the needed data. In FILA, the transmission of data is discrete.

In POT [4] protocol, they classify sensor nodes into a number of Partial Ordered Trees (POT). The root of the POT makes local ranking lists (LR), and maintains the number $h$ of sensor nodes in POT which were the highest sensor readings, and reports them to the base station. When the base station receives the local LR, it will do global top-k evaluation to confirm the validation of global ranking list (GR). GR contains the k highest readings currently in the networks. POT protocol is useful for the occasion where top-k results are correlated spatially. When the top-k results are not correlated spatially, the FILA outperforms POT.

In PRIM [5] and PRIM-c [6] protocols, the TDMA schedule is divided into N partial TDMA frames for collecting sensor readings. The sensed readings are sent at different frames based on different values. The higher sensor readings can be sent to the sink in the more previous frames. The readings maybe sent at multiple frames, which is a typical method that saves energy at the expense of time. It will prolong the querying time and is not suitable for the continuous querying with short interval between successive queries. In PRIM [5], a routing tree is established using conventional TAG-based algorithms. In PRIM-c [6], a routing tree is established using the hierarchical clustering and routing algorithms, such as LEACH, HEED, etc,.

Second, we will show some classical routing protocols. In general, the routing protocols are classified into two types for WSNs: flat routing protocols and hierarchical routing protocols. In flat routing protocols, they route data to the sink node through a multi-hop network. In hierarchical routing protocols, sensor nodes construct clusters for routing, and then data transmission occurs as two steps, i.e., intra-cluster routing and inter-cluster routing [10].

### 2.1. Flat routing protocols

TAG [7] is a tree-based routing approach. First, sink broadcasts a message to build the tree. The node which received the message will be the child of the sender, and then the child node will transfer the building message to its neighbors, and so on. As mentioned above, in TAG, the node close to the sink will die fast, and shorten the network lifetime.

In HEAR protocol, the authors of [11] analyzed theoretically the relationship between the energy consumption and hops, and proposed the actual selection criteria of the number of sub-optimal hops. For example if the distance between node and node is less than 104 m, using direct communications method will be more energy saving; if the distance between node and node is greater than 104 m and less than 2d0 m, using two-hop communication method will be more energy saving, and so on. In HEAR protocol, the source node decides the transmission method based on the actual selection criteria. If one node selects a multi-hop manner, then it will calculate a parameter *n* of the optimal number of hops. If it finds a suitable neighbor node in the [d/n, d/n+ Δ] rang, then it will send a RREQ message to the neighbor node. If the neighbor node receives the RREQ message, then it will send an ACK response message, and try to find its own next hop node, and so on, until it reaches the sink node. Finally, the sink node will send back a RREP message along the relay links.

### 2.2. Hierarchical routing protocols

LEACH [8] is a typical clustering routing algorithm for WSNs. There are two phases in each round including set-up and steady-state phases. In the set-up phase, the cluster heads are randomly selected. After clustering, member nodes select a suitable cluster head to join in. In the steady-state phase, each member node sends sensed data to its cluster head. Cluster head collects and aggregates the incoming data of its member node, and sends them to the sink. However, LEACH cannot guarantee that the clusters are evenly distributed.

In EEUC [12] and UCR [13] protocol, each tentative cluster head has a competition range *Rcomp*. The closer to the sink, the smaller the competition range the tentative cluster head has. Finally, the sizes of clusters would be unequal. However, when the sensor nodes are randomly deployed in the monitoring region, a large number of sensor nodes maybe be deployed closed to the sink. Though the competition ranges *Rcomp*of tentative cluster heads are small, the number of nodes within this region is large. When the cluster heads forward data, the energy consumption will be large too, and so they will die much faster than the other cluster heads, thus shortening the network lifetime.

HEED [9] is another distributed clustering algorithm. The probability of becoming the temporary cluster head is based on the residual energy and communication costs. In the clustering phase, the distribution of nodes within clusters is not taken into account. This may cause uneven distribution of nodes within clusters.

EB-UCP [14] uses an unequal clustering algorithm to balance the energy consumption. The closer the clusters to the sink, the smaller the sizes that the clusters have. In EB-UCP, all nodes need to be pre-deployed in a circular area, and only the sink is deployed at the center point of the area.

## 3. Preliminaries

In our consideration, *N* sensor nodes are cast by plane or other modes in an area. These nodes constitute a network in a self-organized manner, and sample the data periodically. User continuously requests the list of *k* sensor nodes with the highest (or lowest) readings at every sampling period. The *i*th sensor node is denoted by *si*, and the corresponding sensor nodes set $S = \{s1, s2,..., sn\}$, $|S| = N$. Each *si* has the maximum communication radius *R*. $d(si, sj)$ or $d_{ij}$ denotes the communication distance between *si* and *sj*. $d_{ik}$ denotes the communication distance between *si* and the sink.

We make the following assumptions:

1. All ordinary nodes are homogeneous and have the same capabilities. When all nodes are deployed, they will be stationary, and each one has a unique identification (ID).
2. There is only one sink (base station), and the sink node can be recharged.
3. At general top-k querying algorithm, a node may need to know its exact location. However, in our scheme, even though the nodes do not know their exact locations, such as not being equipped with GPS, our clustering algorithm can perform the same.
4. All ordinary nodes have the data fusion capabilities. For top-k querying, multiple data packets can be compressed into one same-size packet. All ordinary nodes are capable of operating in both the active and low-power sleeping modes.
5. Links are symmetric. If the transmission power of the sender is known, the receiver can calculate the

approximate distance based on the received signal strength.

6. The energy resource of ordinary sensor nodes is limited and unreplenished, and sensor nodes can adjust the transmission power based on the distance between the sender and the receiver.

In this article, we use the same model of wireless communication as shown in [10-13,15-18]. When the distance between the sender and the receiver is lower than the threshold value of d0, our scheme will use the free space model ($d^2$ power loss), otherwise use the multi-path fading model ($d^4$ power loss).

The energy consumed for transmission of an $l$-bit packet over distance $d$ is

$$E_{\mathrm{TX}}(l,d) = lE_{\mathrm{elec}} + l_{\in}d^{\alpha} = \begin{cases} lE_{\mathrm{elec}} + l_{\in fs}d^2, d < d0 \\ lE_{\mathrm{elec}} + l_{\in mp}d^4, d \geq d0 \end{cases} \quad (1)$$

Where $Eelec$ denotes the electronics energy. $_{\in fs}d^2$ and $_{\in mp}d^4$ are the energy consumed by power amplifiers. $d0$ is a constant. To receive an $l$-bit packet, the receiver consumes energy as follows:

$$E_{RX}(l) = lE_{\mathrm{elec}} \quad (2)$$

Because the readings collected by adjacent nodes are highly correlated, data aggregation (fusion) can be used [19]. Data aggregation also consumes $EDA$ (nJ/bit/signal) amount of energy.

## 4. CRTQ scheme

Our scheme CRTQ is defined as the order-sensitive exact top-k set querying of FILA based on our clustering routing algorithm. Hence in this section, we will introduce the clustering-based routing algorithm in detail. The routing algorithm consists of two parts; one is the cluster formation algorithm; and the other part is the inter-cluster choosing the relay cluster head algorithm.

During the cluster formation algorithm, sender sensors keep their fixed maximum transmission power. Hence, a receiver node can compute the approximate distance to sender node based on the received signal strength.

In the network deployment phase, the sink broadcasts a beacon message at a fixed power to all the nodes in the network. Hence each node can calculate the approximate distance $d_{ik}$ according to the received signal strength, in which $d_{ik}$ will be used in the inter-cluster choosing the relay cluster head algorithm.

In the following, we describe CRTQ in detail:

### 4.1. Cluster formation

In the cluster formation algorithm[a], each node uses a self-organized and competition-based algorithm to form clusters. Each cluster has a unique cluster head, and ordinary member nodes use direct communication link to communicate with cluster head. The competition range of node decreases as its residual energy decreases to lower than the average residual energy of all nodes in the network. The result is that cluster heads with less residual energy have smaller cluster sizes, thus consuming lower energy during the intra-cluster communication, and being able to save more energy for the inter-cluster communication.

**Procedure 1:** CF procedure

```
1:Procedure begin/* $\forall si \in V$ */
2:u $\Re$ RAND(0, 1)
3:if u < T
    4: $t = k * \frac{E_{ini}-E_{res}}{E_{ini}}$;
    5: schedule a FINAL_HEAD_MSG($ID, R_c, E_{res}, d_{ik}$) message with $t$ delay time;
6:else
    7: exit;
8:end-if
9:if any FINAL_HEAD_MSG($ID, R_c, E_{res}, d_{jk}$) from $sj$ is overheard within $t$ delay time
    10: if $d$ ($si$, $sj$ ) < $sj.Rc$
        11: cancel the scheduled FINAL_HEAD_MSG($ID, R_c, E_{res}, d_{ik}$) message;
        12: exit;
    13: else
        14: waiting and overhearing;
    15: end-if
16:else
    17: proceed the scheduled FINAL_HEAD_MSG($ID, R_c, E_{res}, d_{ik}$) message;
18:end-if
```

Cluster formation (CF) procedure is presented in Procedure 1. Here, we introduce the clustering algorithm in detail.

Similar to LEACH and EEUC, the operation of CRTQ is divided into rounds. In each round, first, all ordinary nodes compete for tentative cluster heads with probability $T$, in which $T$ is the default threshold. Ordinary nodes that fail to be tentative heads keep overhearing until the clustering ends.

A CF procedure is shown in Procedure 1, where $t$ is a delay function; $E_{\mathrm{res}}$ denotes the residual energy of tentative cluster head; $E_{\mathrm{ini}}$ denotes the initial energy of tentative cluster head; $k$ is a constant real number, as a parameter to adjust the length of time $t$. We assume that all sensors are homogeneous and have the same $E_{\mathrm{ini}}$. When the timer $t$ expires, the tentative cluster head will broadcast a *FINAL_HEAD_MSG*($ID$, $R_c$, $E_{res}$, $d_{ik}$) message and compete successfully as a final cluster head. Tentative cluster heads with more residual energy have smaller t, and have more probabilities to compete successfully as final cluster heads (lines 2-8).

Each tentative cluster head has a competition range $R_c$, and different $R_c$ ranges represent different cluster sizes. In each scope of $R_c$, there is only one cluster head. The competition range $R_c$ of tentative cluster head $s_i$ can be expressed as a linear function of its residual energy and the average residual of all nodes:

$$R_c = (\frac{E_{\text{res}} - E_{\text{avg}}}{E_{\text{avg}}}) * R_e + R_0 \quad (3)$$

where $R_0$ is the competition range which is predefined, and $E_{\text{avg}}$ denotes the average residual energy value of all nodes in the network. The sink will compute the average residual energy value of all nodes and broadcast the $E_{\text{avg}}$ message in the network area. How are we to obtain the value of $E_{\text{avg}}$? We will describe it in Section 4.3. *Re* denotes the competition range affected by residual energy. As shown in Equation (3), $R_c$ varies from $R_0$ - $R_e$ to $R_0$ + $k$ * $R_e$ (in which, $k$ is proportional to $E_{\text{res}}$ and $E_{\text{avg}}$) according to the margin between the residual energy of tentative cluster head and the average residual energy of all nodes. If the competition radius *Rc* is too small (e.g., $R_c$ is less than 1 m), then there will be a lot of single-node clusters, and it will affect the performance of the algorithm. Hence we let $R_c$ is greater than 1 m to avoid this problem.

Cluster heads closer to the base station have higher load of relay traffic in the inter-cluster communication, and so they will consume more energy. However, in the subsequent rounds, their $R_c$ will decrease, and less member nodes will belong to the cluster head. The cluster heads will reduce the intra-cluster communication energy consumption and counteract the increased inter-cluster communication energy consumption. This method will balance the energy consumption among all the cluster heads, avoid the hot spot problem [12], and extend the network lifetime.

If any *FINAL_HEAD_MSG*(*ID*, $R_c$, $E_{\text{res}}$, $d_{jk}$) message from *sj* is overheard by the tentative cluster head or the final cluster head *si*, *si* will compute the distance *d* (*si, sj* ) between *si* and *sj*, and then save (*ID*, $R_c$, $E_{\text{res}}$, $d_{jk}$) of *sj* and d(*si, sj* ) in its cache, and add *sj* to *Nsi*, which may be used for inter-cluster choosing the relay cluster head algorithm. In Procedure 1, lines 9-18, if any *FINAL_HEAD_MSG* message from *sj* is overheard by *si* before the timer *t* expires, the tentative cluster head node *si* will compute the distance *d*(*si, sj* ) between *si* and *sj* . If *d*(*si* , *sj*) < *sj*.$R_c$, then *si* will give up the competition and cancel the scheduled *FINAL_HEAD_MSG* message immediately, and exit as an ordinary node. Otherwise, the tentative cluster head node will continue waiting and overhearing. If the timer *t* expires, then *si* will proceed the scheduled *FINAL_HEAD_MSG* message.

After all final cluster heads have been selected, each ordinary node chooses the closest cluster head with the strongest received signal in its $N_{si}$ and sends a *JOIN_REQ_MES* (*ID*, *d*(*sj*, *ck*), $E_{\text{res}}$, *CHID*) message. *d*(*sj*, *ck*) denotes the distance between *sj* and the corresponding cluster head *ck*. *CHID* denotes cluster head ID that *sj* belongs to. The cluster head can get the maximum *d*(*sj* , *ck*) from its member nodes. When it is the time to broadcast a message to its member nodes, the transmission radius may be the maximum *d*(*sj*, *ck*).

Figure 2 gives an overview of the CRTQ protocol after clustering, where the unequal cells represent the unequal clusters formed by CRTQ, and *CH* denotes cluster head.

### 4.2. Inter-cluster choosing the relay cluster head

In the inter-cluster choosing the relay cluster head algorithm, each cluster head tries to find a neighbor cluster head which is closer to the sink (the base station) as a relay node. The relay node looks for another near-neighbor cluster head as its relay node, until to the sink. Finally, it will form one or more data relay links. Sensed data are aggregated along these data relay links. In this way, the distance of relay link is shorter than direct link usually, and hence, the cluster heads can reduce the energy consumption.

**Procedure 2:** Choosing the relay cluster head procedure

```
1:/* $N_{si}$ is the neighbor cluster heads set of $si$*/
2: $W_{si} = N_{si}$
3:while(monitoring)
4: if $si$ is the farthest node of $W_{si}$ or $si$ is the only node of $W_{si}$ or timeout
```

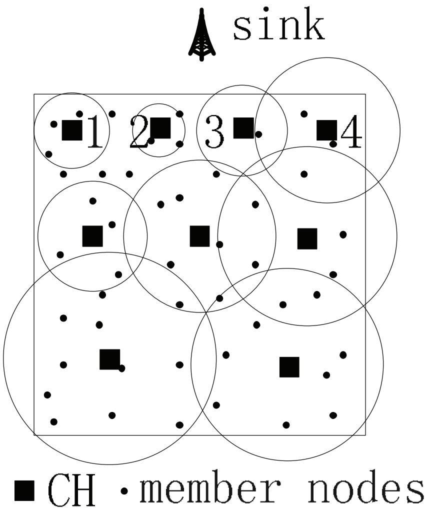


**Figure 2 An overview of the CRTQ protocol**. © August 18, 2011 AED: lastpage setting LastPage.

```
5: while $sj \in N_{si}$ and $d_{jk} < d_{ik}$ and $d_{ij} < d_{ik}$ and $cost_{ik} > (cost_{ij} + cost_{jk})$
6:     $pj = \theta \frac{d_{ik}-d_{ij}}{d_{ik}} + (1-\theta)\frac{E_{j-res}}{E_{j-ini}} (0 \le \theta \le 1)$
7:     add $sj$ to $rp_{si}$-set;
8: end-while
9: find the max $pj$ in $rp_{si}$-set and the node $j$ as the data relay point $rp_{si}$;
10: if $rp_{si}$ exists
11:     send $m^{si}_{rpsi}(si.ID, rp_{si}.ID)$ to $rp_{si}$;
12: else /* $rp_{si}$ does not exist */
13:     call the inter-cluster probe procedudure;
14: end-if
15: break;/* exit the while monitoring iteration */
16: end-if
17: if $m^{sj}_{rpsj}(sj.ID, rp_{sj}.ID)$ is received
18:     if $sj \in W_{si}$
19:         delete the $sj$ node from $W_{si}$ set;
20:     end-if
21:     if $rp_{sj}$. $ID$ = $si$. $ID$/*the relay point of $sj$ is $si$*/
22:         $si$ save the information of child node $sj$;
23:     end-if
24: end-if
25:end-while
```

At the beginning of the algorithm, as shown in Procedure 2, the neighbor cluster head set of cluster head *si* is denoted by $N_{si}$ , which is obtained at cluster formation phase. $W_{si}$ denotes the temporary neighbor set of *si*. Initially, the $W_{si}$ equals $N_{si}$. In the process of choosing the relay node, all cluster heads are monitoring. As shown in Procedure 2, lines 17-24, if a $m^{sj}_{rpsj}(sj.ID, rp_{sj}.ID)$ message is received by *si*, and the sender of the message belongs to the $W_{si}$, *si* will delete the *sj* node from $W_{si}$ set. If the receiver of the message is *si* itself, then it means that the *si* is the relay cluster head of *sj*, and *si* will save the information of child node *sj*. This information will be used to decide to whether aggregate and transmit the received data message in the steady phase. Only if the sender of the sensed data message is one of the children of *si*, *si* will aggregate and transmit the received message.

In monitoring, if *si* is the farthest node of $W_{si}$, or *si* is the only node of $W_{si}$, or the time of choosing the relay node is expired, then the cluster head *si* will initialize the process of choosing the relay node. In our scheme, the process of choosing the relay node is divided into two phases. First, we use the information of neighbor cluster head set, which is obtained in the CF phase, to find out the relay cluster head node. If the suitable relay node does not exist, then we will use the second algorithm which is a probe-based algorithm to find out the relay cluster head node. As shown in Procedure 2, lines 4-16, $d_{ij}$ denotes the communication distance between *si* and *sj*; $d_{ik}$ denotes the communication distance between *si* and the sink node; and $d_{jk}$ denotes the communication distance between *sj* and the sink node. Such information is obtained in the CF phase. cost$_{ik}$ denotes the energy consumed by *si* when *si* sends a sensed data message to the sink node; cost$_{ij}$ denotes the energy consumed by *si* when *si* sends a sensed data message to *sj*; cost$_{jk}$ denotes the energy consumed by *sj* when *sj* sends a sensed data message to the sink node. When it is time to compute the energy consumption, we assume that both the free space ($d^2$ power loss) propagation channel model and the multi-path fading ($d^4$ power loss) channel model are used, according to the distance between the transmitter and receiver.

The tentative relay point set $rp_{si}$-set of *si* satisfies the conditions: $sj \in N_{si}$, $d_{jk} < d_{ik}$, $d_{ij} < d_{ik}$ and cost$_{ik}$ > (cost$_{ij}$ + cost$_{jk}$ ). As shown in Figure 3, the shadow part maybe the $rp_{si}$-set of *si*.

Every node in $rp_{si}$-set computes the *pj*, where E$_{j\text{-res}}$ denotes the residual energy of node *sj*; E$_{j\text{-ini}}$ denotes the initial energy of node *sj*; $\theta$ is a constant coefficient between 0 and 1. Cluster head *si* selects another cluster head *sj* which has the maximum *pj* as the relay point $rp_{si}$.

If $rp_{si}$ exists, *si* will send a $m^{si}_{rpsi}(si.ID, rp_{si}.ID)$ message to the relay cluster head $rp_{si}$ and exit. Otherwise, *si* call the inter-cluster probe procedure to find the relay node. The inter-cluster probe procedure is a probe-based algorithm. We will describe the algorithm in detail.

**Procedure 3:** Inter-cluster probe procedure

```
1:/*inter-cluster probe procedure*/
2:if $d_{ik} < TD\_MAX$
3:     send $m^{si}_{k}(si.ID, k.ID)$ to sink node $k$;
4: else
5:     DisTemp = MAX_DISTANCE + d0;
```

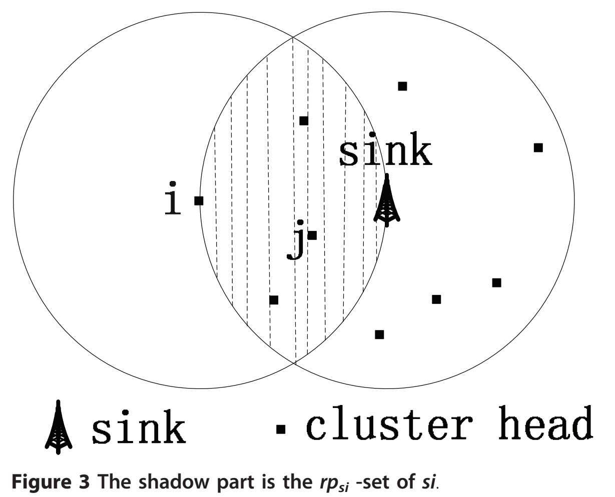


**Figure 3 The shadow part is the** ***rp*$_{si}$ -set of *si*.**

```
    FindRelayPointFlag = FALSE;
    while DisTemp < $d_{ik}$
        while $d_{jk} < d_{ik}$ and $d_{ij} < DisTemp$ and $cost_{ik} > (cost_{ij} + cost_{jk})$
            $p_j = \theta \frac{d_{ik}-d_{ij}}{d_{ik}} + (1-\theta)\frac{E_{j-res}}{E_{j-ini}} \quad (0 \le \theta \le 1)$
            add $sj$ to $rp_{si}$ -set;
        end-while
        find the max $pj$ in $rp_{si}$ -set and the node $j$ as the data relay point $rp_{si}$;
        if $rp_{si}$ exists
            send $m_{rpsi}^{si}(si.ID, rp_{si}.ID)$ to $rp_{si}$;
            FindRelayPointFlag = TRUE;
            break;/*exit the while DisTemp < $d_{ik}$ */
        else
            DisTemp = DisTemp + d0;
        end-if
    end-while
    if FindRelayPointFlag == FALSE
        send $m_{k}^{si}(si.ID, k.ID)$ to sink node $k$;
    end-if
end-if
```

In the inter-cluster probe procedure, as shown in Procedure 3. *TD_MAX* is a predefined constant distance to the sink node. During the intra-cluster communication part, it is considered that sensors keep their fixed maximum transmission power, which usually means the fixed maximum transmission distance. Here, we use the *MAX_DISTANCE* which denotes the fixed maximum transmission distance.

If $d_{ik} < TD\_MAX$, the cluster head *si* will send the sensed data to sink directly. During the intracluster communication part, the cluster head *si* can overhear the broadcast message of neighbor cluster head within *MAX_DISTANCE* distance. Hence, in our probe-based algorithm, the probe distance is equal or greater than (*MAX_DISTANCE* + d0) distance, in which d0 is the threshold distance of Equation 1. If there is a suitable relay cluster head, *si* will send a $m_{rpsi}^{si}(si.ID, rp_{si}.ID)$ message to the relay cluster head $rp_{si}$ and exit, otherwise, the probe distance will be increased by d0. If the probe distance is equal or greater than $d_{ik}$, and *si* still has not find a suitable relay cluster head, *si* will send the sensed data to sink directly.

When the clustering is completed, the steady-state operation (data transmission or top-k query) starts. When a relay cluster head receives the sensed data from its neighbor, the relay node will aggregate them with its own sensed data into one same-size packet and forward to sink along the data relay link.

### 4.3. Clustering time

In the hierarchical routing protocols, such as EEUC [12] and UCR [13], the clustering time is static. One round includes five sampling periods, and the clustering time is the time of one round (five sampling periods). If the cluster heads have enough energy and the clustering timer expires, then there is a clustering process in the whole network, which will waste energy consumption. If the energy of cluster heads has been significantly lower than the average energy of network, then the delayed clustering will exhaust energy of parts of cluster heads faster and destroy the connectivity of network. Therefore, the static clustering time is not good enough to balance energy consumption among sensor nodes.

In our scheme, we propose a dynamic clustering algorithm. Our routing scheme is used for top-k querying. In every sampling period, there is only a part of the sensor nodes. which need to send sensed data to the sink. To get the energy distribution information of the network, the value of residual energy will be sent to the sink node along with the sensed data message. Regardless of the source nodes transmitting messages to the sink node, or intermediate nodes forwarding the messages of other nodes to the sink node, these messages will carry the values of residual energy of the source nodes or intermediate nodes.

During every sampling period(round), if a node *si* does not send a message to sink, the sink node can use the value of residual energy of *si* which is transmitted latest and calculate the present value of residual energy of *si* according to the history of energy consumption pattern to calculate the average residual energy of all nodes or part of nodes. For example, the value of residual energy of a node *si* is 1.0 joule at the *i*th round. At the ($i$ + 1) th round, the readings of the node *si* are very low, and so it does not send a message to the sink node. The sink node when it does not receive a message with the value of residual energy will think the node *si* is in sleeping mode at the ($i$ + 1)th round. Assuming that the energy consumed in sleeping mode is 0.00001 joule, which is a constant generally, the sink node will compute the value of residual energy of the node *si* which is 0.99999 joule (1.0-0.00001) at the ($i$ + 1)th round.

**Procedure 4:** Computing the average residual energy of all cluster heads

```
NumberOfCh = 0;/*count the number of all cluster heads*/
SumChEnergy = 0;/*calculate the sum energy of all cluster heads*/
for i = 1 to NumOfNodes/*NumOfNodes means the number of nodes in WSNs*/
    if $si$ is cluster head
        if $si$ transmitted the message to sink
            SumChEnergy = SumChEnergy+$si.CurEn$
        else
            SumChEnergy = SumChEnergy+ $si.CalEn$
        end-if
        NumberOfCh = NumberOfCh+1;
    end-if
end-for
if NumberOfCh > 0
    AvrChEn = SumChEnergy/NumberOfCh;
else
    AvrChEn = 0;
end-if
```

As shown in Procedure 4, *si.CurEn* means that *si* sends a sensed data message along with energy value to the sink in current sampling period (round). *si.CalEn* means that *si* does not send a message to the sink, and so uses the calculated value according to the latest history value to count. *AvrChEn* means the average residual energy of all cluster heads. Using the same method, we can get the average value of residual energy of all non-cluster head nodes *AvrNonChEn* in the WSNs. In sampling period (round), after the sink obtained the necessary information and completed the top-k computation, the sink would compare the residual energy value of every cluster head with the average residual energy of all non-cluster heads which is multiplied by a constant coefficient *S_FACTOR*. Sink may compare the average residual energy of all cluster heads with the average residual energy of all non-cluster heads which multiplied by a constant coefficient *G_FACTOR*. As shown in Procedure 5, *S_FACTOR* and *G_FACTOR* are constant coefficients; *AvrNonChEn* means the average residual energy of all non-cluster heads. If *si* sends a message to the sink at current sampling period, then *si.En* is *si.CurEn* , otherwise *si.En* is *si. CalEn* . If *AdjustFlag = True*, sink will broadcast a message to all nodes in the WSNs to notify clustering, otherwise latest clustering results will be retained.

**Procedure 5:** Comparing the residual energy values

```
AdjustFlag = FALSE;
for every cluster head si
    if si.En < (AvrNonChEn*S_FACTOR)
        AdjustFlag = TRUE;
        break;
    end-if
end-for
if AdjustFlag == FALSE
    if AvrChEn < (AvrNonChEn * G_FACTOR)
        AdjustFlag = TRUE;
    end-if
end-if
```

## 5. Simulation results

To analyze the performance of CRTQ algorithm, we conduct experiments using an MATLAB implementation. We use two kinds of datasets. One is the real dataset, which can be downloaded from http://www.cs.cmu.edu/afs/cs/project/spirit-1/www/#Download. The dataset is produced by Jimeng Sun and Christos Faloutsos (CMU,Carnegie Mellon University). The original data are provided by Carlos Guestrin (CMU). We use the q8calibHumTemp.dat file which includes temperature records. The file includes 56 records in every column, which are collected from sensors. Another dataset is a synthetic dataset, which is generated using Gaussian distribution in which mean is 20 and the standard deviation is 0.7. This data file includes 300 records each column.

The sensor nodes can operate in one of three modes: sending message, receiving message, and sleeping. The energy consumption is different in these modes. As shown in Table 1, we can see part of the MICA2 http://www.xbow.com.cn/wsn/pdf/MICA2.pdf parameters. The percent of current drawn in sleeping mode to current drawn in receiving mode is (15+1)$\mu$A/(8+10)mA = 0.089%. As shown in Table 2[20], The percentage of power consumption in sleep mode to receiving power is 0.016 mw/13 mw = 0.123%. The current drawn or the power consumption in the sleeping mode is very little. Hence in our simulation, we omit the power consumption in the sleeping mode.

We assume the communication links are error free, and MAC layer is ideal. To compute the network lifetime, we define a sampling period as a round. Each of the following simulation results represents an average summary of 20 runs.

The transmission range of sink node is usually greater than the transmission range of ordinary sensor nodes, and so we assume that the transmission range of sink node can cover most regions of the monitoring networks.

### 5.1. Parameter setting

CRTQ protocol involves many parameters, such as $T$, $R0$, $Re$, $\theta$ , $S\_FACTOR$ and $G\_FACTOR$. We try to obtain optimal parameters to extend the network lifetime. We use the number of rounds until the first node dies to describe the network lifetime. We randomly deployed 56 homogeneous sensor nodes in the 200*200$m^2$ rectangular region, and the sink is located at the center. Table 3 shows the relevant parameters, in

**Table 1 Part of MICA2 parameters**

| Processor performance | | |
|---|---|---|
| Current draw | 8 mA | Active mode |
| | <15$\mu$A | Sleep mode |
| Multi-channel radio | | |
| Current draw | 27 mA | Transmit with maximum power |
| | 10 mA | Receive |
| | <1$\mu$A | Sleep |

**Table 2 Parameters**

| Parameters | Values |
|---|---|
| Transmission power | 14 mW |
| Receiving power | 13 mW |
| Power consumption in sleep mode | 0.016 mW |

which the parameters of radio model are same as those in EEUC. The dataset is the real dataset.

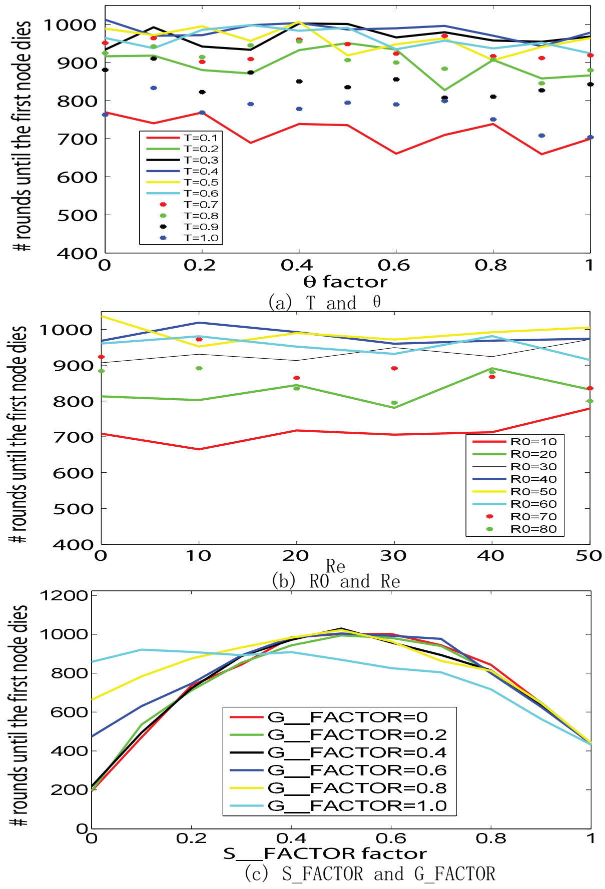
First, we observe the network lifetime affected by the parameter $T$ (Procedure 1) and $\theta$ (Procedure 2). Figure 4a shows the relationship between $T$ and the network lifetime, and the relationship between $\theta$ and the network lifetime. $T$ increases from 0.1 to 1.0, and $\theta$ increases from 0 to 1.0. The number of rounds until the first node dies change roughly from 650 to 1050. $T$ is the major factor that affects the network lifetime. There is an optimal range for the value of $T$, i.e., 0.3-0.6. We need to choose the optimal parameters of $T$ and $\theta$ to ensure distributed clusters, and to extend the network lifetime.

Second, we look at the impact of parameters $R0$ and $Re$ (in Equation 3) on the network lifetime. Figure 4b shows the relationship between the network lifetime with parameters $R0$ and $Re$. We can see from the figure that the $R0$ is the major factor which affects the network lifetime. Because $R0$ mainly determines both the $Rc$ (in Equation 3) and the number of clusters in a given network, there is an optimal range for the value of $R0$, i.e., 40-60 m. $Re$ determines the difference of cluster sizes. When $R0$ is fixed, the network lifetime varies as $Re$ varies. Therefore, there exists optimal values of $R0$ and $Re$, which could best prolong the network lifetime.

Finally, we investigate the impacts of *S_FACTOR* and *G_FACTOR* (Procedure 5) on the network lifetime. Figure 4c shows that the *S_FACTOR* is the major factor which affects the network lifetime. There is an optimal range for the value of *S_FACTOR*, i.e., 0.4-0.6. If *S_FACTOR* is set to 0, even though the residual energy value of one of the cluster heads is very small, the sink will not broadcast a message to notify clustering. It will shorten the network lifetime. As *S_FACTOR* increases from 0, the network lifetime has also increased, and the energy consumption becomes gradually balanced. However, if *S_FACTOR* is too large, the lifetime decreases, because there are too many clustering frequencies. Therefore, there exists optimal values of *S_FACTOR* and *G_FACTOR*, which could best extend the network lifetime.

**Table 3 Simulation parameters**

| Parameter | Value |
|---|---|
| Network field | (0,0)-(200,200) m |
| Base station location | (100,100) m |
| **N** | 56 |
| Initial energy | 0.05 **J** |
| $E_{elec}$ | 50 nJ/bit |
| $E_{fs}$ | 10 pJ/(bit*m2) |
| $E_{mp}$ | 0.0013 pJ/(bit*m4) |
| *EDA* | 5 nJ/(bit*signal) |
| Data packet size | 100 Bytes |
| Broadcast packet size | 25 Bytes |
| Threshold distance (*d*0) | 87 m |



(a) T and θ

(b) R0 and Re

(c) S_FACTOR and G_FACTOR

**Figure 4 The network lifetime with different parameters. (a)** *T* **and** $\theta$**; (b)** *R0* **and** *Re***; (c)** *S_FACTOR* **and** *G_FACTOR*.

In our simulation, we use the comprehensive and optimized parameters. We set $T$ = 0.4, $\theta$ = 0.3, $R0$ = 40 m, $Re$ = 10 m, $S\_FACTOR$ = 0.4 and $G\_FACTOR$ = 0.8.

### 5.2. Performance analysis

We compare our CRTQ algorithm with the following three popular algorithms.

**1.FILA:** FILA is a top-k querying schemes based on TAG tree routing. We use the order-sensitive exact top-k set querying of FILA as comparison.

**2.EEUC-Q:** EEUC is the most similar self-organized clustering protocol to our clustering algorithm. It is a good baseline for comparison, because it has the following characteristics: (1) clustering is based on local information, and cluster heads are well-distributed; (2) sensor nodes do not need to be pre-deployed; and (3) no underlying assumptions about the network are made. We define that using the order-sensitive exact top-k set querying of FILA based on EEUC routing as EEUC-Q.

**3.HEAR-Q:** HEAR is a flat routing algorithm, but the performance of network lifetime is better than the clustering routing algorithms LEACH [8] and HEED [9] as described by the respective authors. We define that using the order-sensitive exact top-k set querying of FILA based on HEAR routing as HEAR-Q.

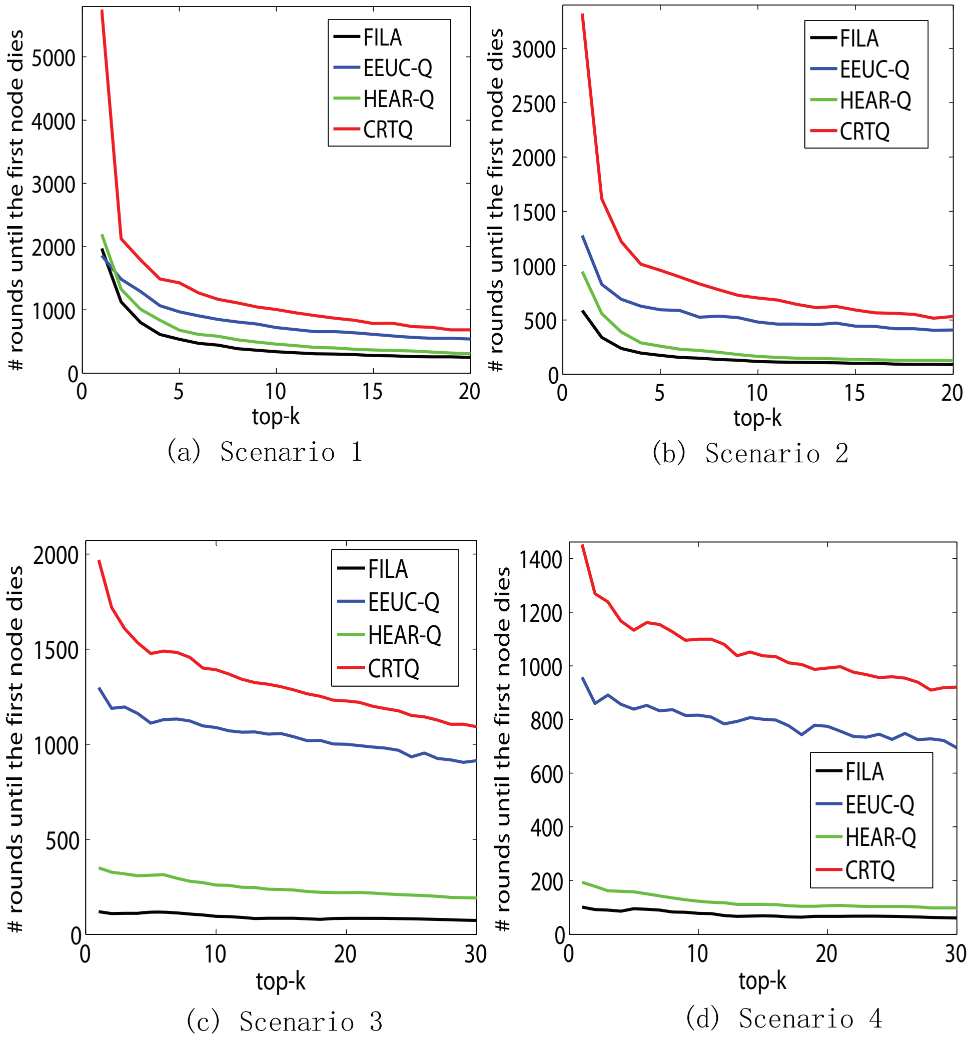
The parameters of radio model are shown in Table 3. In EEUC, one round includes five TDMA frames, and one TDMA frame means one sampling period, and one round means one clustering. Hence in EEUC-Q, there is a clustering for every five sampling periods (we define a sampling period as a round in this article). For EEUC-Q, in our experiment, the optimal parameters are $T$ = 0.4, $Rc$ 0 = 40, $C$ = 0.3 in the intra-cluster communication. CRTQ and EEUC-Q use the same optimal parameters $TD\_MAX$= 30 in the inter-cluster communication. The transmission radii of FILA and HEAR-Q are set to be 104 m.

Now, we consider the following four scenarios:

**Scenario 1:** We randomly deployed 56 homogeneous sensor nodes in the 200*200 $m^2$ rectangular region, and the sink was located at the center (100 m, 100 m). In this scenario, we use the real dataset.

**Scenario 2:** We randomly deployed 56 homogeneous sensor nodes in the 200*200 $m^2$ rectangular region, and the sink was located at the outside of the region, (100 m, 250 m). In this scenario, we use the real dataset.

**Scenario 3:** We randomly deployed 300 homogeneous sensor nodes in the 300*300 $m^2$ rectangular region, and



(a) Scenario 1

(b) Scenario 2

(c) Scenario 3

(d) Scenario 4

**Figure 5 The network lifetime with different top-k querying**. (a)Scenario 1; (b)Scenario 2; (c)Scenario 3; (d)Scenario 4.

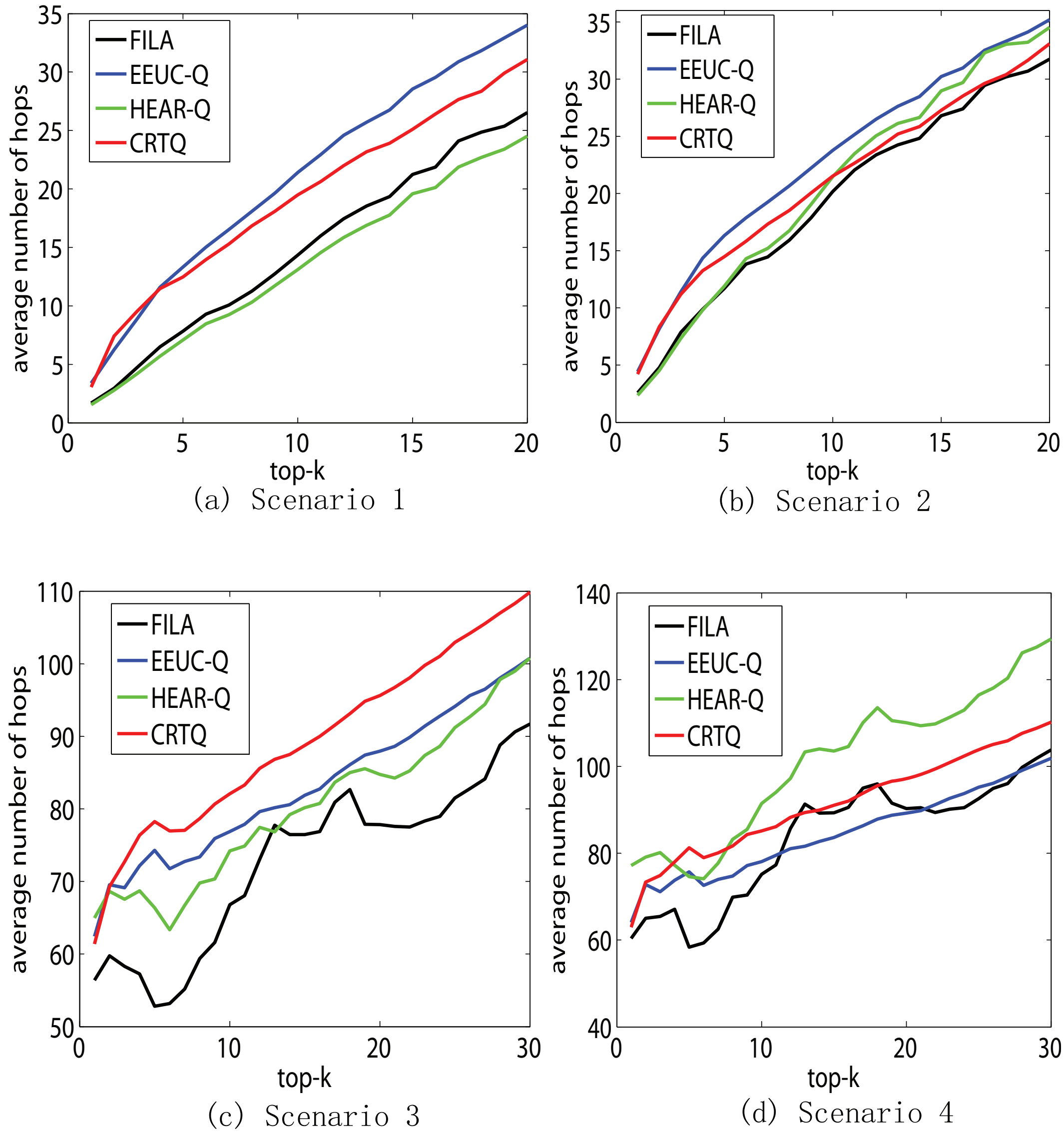


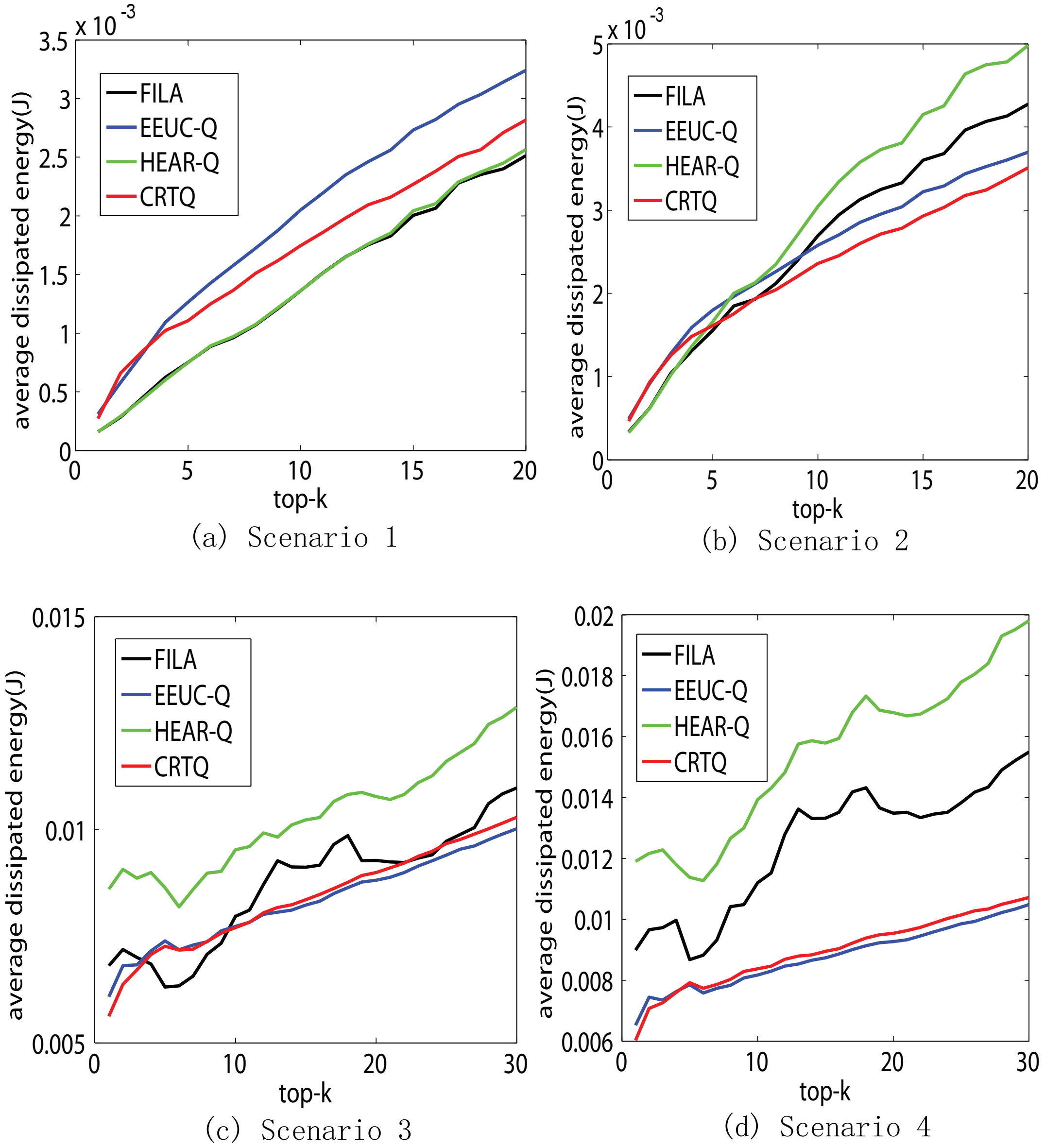
**Figure 6 The average number of hops with different top-k querying**. (a)Scenario 1; (b)Scenario 2; (c)Scenario 3; (d)Scenario 4.

the sink was located at the center (150 m, 150 m). In this scenario, we use the synthetic dataset.

**Scenario 4:** We randomly deployed 300 homogeneous sensor nodes in the 300*300 $m^2$ rectangular region, and the sink was located at the outside of the region (150 m, 350 m). In this scenario, we use the synthetic dataset.

First, we compare the network lifetimes of CRTQ, FILA, EEUC-Q and HEAR-Q in different scenarios. Figure 5 shows the number of sensor nodes still alive changing over different top-k queryings. Top-k varies from top-1 to top-20 or top-30. HEAR-Q is based on HEAR routing algorithm, and FILA is based on TAG routing algorithm. Both HEAR and TAG are flat routing algorithms. In HEAR protocol, the node tries to find an optimal and suitable relay node, whenever in TAG protocol, all nodes only try to find their neighbors. Hence HEAR-Q extends the network lifetime over FILA. CRTQ and EEUC-Q are based on clustering routing algorithms. There is additional clustering overhead, such as control messages and energy consumption in clustering routing, but the clustering routing algorithms can balance the load distribution among all sensor nodes. Hence CRTQ and EEUC-Q extend the network lifetimes over FILA and HEAR-Q. CRTQ takes more consideration on the balanced distribution of load of all nodes, and so CRTQ clearly extends the network lifetimes over FILA, EEUC-Q, and HEAR-Q in all scenarios. In scenario 2, the sink is located at the outside of the monitoring region. The average distance between sensor node and the sink is longer than the distance of scenario 1. The network lifetime of scenario 2 is shorter than scenario 1 at the same top-k querying, as well as scenarios 3 and 4.

Second, we compare the average number of hops and average dissipated energy each round of CRTQ, FILA, EEUC-Q, and HEAR-Q in different scenarios over different top-k queryings. Figure 6 shows the average number of hops every round over different top-k queryings, and Figure 7 shows the average dissipated energy every round over different top-k queryings. The scenarios 1 and 2 are the occasions where a small number of nodes are deployed in a small region. In scenario 1, the average number of hops and average dissipated energy of clustering routing algorithms CRTQ and EEUC-Q are greater than flat routing algorithms FILA and HEAR-Q, but the clustering routing algorithms can balance the load distribution, and CRTQ and EEUC-Q extend the network lifetimes over FILA and HEAR-Q. In scenario 2, the average number of hops of CRTQ, EEUC-Q, FILA, and HEAR-Q are almost the same, and the average dissipated energy of CRTQ and EEUC-Q may be less than FILA and HEAR-Q. The scenarios 3 and 4 are the occasions where a large number of nodes are deployed in a large region. In scenario 3, the average numbers of hops of CRTQ and EEUC-Q are almost greater than FILA and HEAR-Q, and the average



(a) Scenario 1

(b) Scenario 2

(c) Scenario 3

(d) Scenario 4

**Figure 7 The average dissipated energy with different top-k querying**. (a)Scenario 1; (b)Scenario 2; (c)Scenario 3; (d)Scenario 4.

dissipated energies of CRTQ and EEUC-Q are almost less than FILA and HEAR-Q. If there is the same distance from one node to the sink node, the fewer the number of forwarding hops, the longer the distance of each hop, and more likely to consume the more energy. In scenario 4, the average numbers of hops of CRTQ, EEUC-Q, FILA, and HEAR-Q are almost the same, and the average dissipated energies of CRTQ and EEUC-Q are less than FILA and HEAR-Q.

As shown in Figure 5, CRTQ clearly extends the network lifetime over EEUC-Q in all scenarios. As shown in Figure 6, the average number of hops each round of CRTQ may be less than EEUC-Q (in scenarios 1 and 2) or greater than EEUC-Q (in scenarios 3 and 4). As shown in Figure 7, the average dissipated energy every round of CRTQ may be less than EEUC-Q (in scenarios 1 and 2) or a little greater than EEUC-Q (in scenarios 3 and 4). Then, we can conclude that balancing the load distribution to extend the network lifetime is more efficient than reducing the number of hops or reducing the dissipated energy each round.

## 6. Conclusions and future work

In this article, we proposed a routing scheme CRTQ for continuous top-k querying in WSNs to save the energy consumption and extend the network lifetime. We introduced the cluster formation algorithm and inter-cluster choosing the relay cluster head algorithm. Moreover, we adopted a corresponding dynamic clustering algorithm. Our experimental results show that the proposed CRTQ scheme can achieve a significant amount of energy savings and can extend the network lifetime.

In the future, we plan to extend the proposed clustering-based approach to other aggregate functions, such as *kNN*, *join*, average, sum, etc.

## Endnote

[a] This section was partly presented at the 1st IET International Conference on Wireless Sensor Network (IET-WSN 2010), "Competition-based Clustering and Energy-saving Data Gathering in Wireless Sensor Networks".

**Acknowledgements**
This research was supported by the National 863 Hi-Tech R&D Plan of China under Grant No. 2008AA01Z120, the National Science Foundation of China (61070056, 61033010), the Important National Science and Technology Specific Projects of China (HGJ Project), and the Research Funds of Renmin University of China (11XNH119, 10XNI018).

**Author details**
[1] Key Laboratory of Data Engineering and Knowledge Engineering, Renmin University of China, MOE, Beijing 100872, China [2] School of Information, Renmin University of China, Beijing 100872, China [3] Hunan University of Science and Technology, Xiangtan 411201, China

**Competing interests**
The authors declare that they have no competing interests.

Received: 24 September 2010 Accepted: 23 August 2011
Published: 23 August 2011

**References**

1. Institute of Electrical and Electronics Engineers Inc, Ten emerging technologies that will change your world. IEEE Eng Manag Rev. **32**, 20–30 (2004)
2. CS Raghavendra, KM Sivalingam, T Zhati, Wireless Sensor Networks (Kluwer Academic Publish-ers, 2004), pp. 185–252
3. M Wu, J Xu, X Tang, W-C Lee, Top-k monitoring in wireless sensor networks. IEEE Trans Knowl Data Eng. **19**(7):962–976 (2007)
4. Y Cho, J Son, YD Chung, POT: an efficient top-k monitoring method for spatially correlated sensor readings, in *5th International Workshop on Data Management for Sensor Networks, DMSN'08, in Conjunction with the 34th International Conference on Very Large Data Bases*,. 8–13 (2008)
5. MH Yeo, DO Seong, JS Yoo, PRIM: priority-based top-k monitoring in wireless sensor networks, in *International Symposium on Computer Science and Its Applications, Proceeding, CSA 2008*,. 326–331 (2008)
6. M Yeo, D Seong, J Yoo, Data-aware top-k monitoring in wireless sensor networks, in *IEEE Radio and Wireless Symposium, Proceedings, RWS 2009*,. 103–106 (2009)
7. S Madden, MJ Franklin, J Hellerstein, W Hong, TAG: a Tiny AGgregation Service for ad-hoc sensor networks, in *Proc Usenix Fifth Symp Operating Systems Design and Implementation (OSDI '02)*,. 131–146 (December 2002)
8. W Heinzelman, A Chandrakasan, H Balakrishnan, Energy efficient communication protocols for wireless microsensor networks, in *Proceedings of the 33rd Hawaiian International Conference on Systems Science*,. 223 (January 2000)
9. O Younis, S Fahmy, HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. IEEE Trans Mobile Comput. **3**(4):660–669 (2004)
10. D-Y Kim, J Cho, B-S Jeong, Practical data transmission in cluster-based sensor networks. KSII Trans Internet Inf Syst. **4**(3):224–242 (2010)
11. J Wang, J Cho, S Lee, K-C Chen, Y-K Lee, Hop-based energy aware routing algorithm for wireless sensor networks. IEICE Trans Commun. **E93B**(2):305–316 (2010)
12. C Li, M Ye, G Chen, J Wu, An energy-efficient unequal clustering mechanism for wireless sensor networks, in *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference* (IEEE Press, Washington, USA, 2005, November 7-10, 2005), pp. 1–8
13. G Chen, C Li, M Ye, J Wu, An unequal cluster-based routing protocol in wireless sensor networks. Wirel Netw. **15**(2):193–207 (2009). doi:10.1007/s11276-007-0035-8
14. J Yang, D Zhang, An energy-balancing unequal clustering protocol for wireless sensor networks. Inf Technol J. **8**(1):57–63 (2009). doi:10.3923/itj.2009.57.63
15. W Heinzelman, Application-specific protocol architectures for wireless networks. Ph.D. Dissertation, Massachusetts Institute of Technology. (June 2000)
16. L Cao, C Xu, W Shao, G Zhang, H Zhou, Q Sun, Y Guo, Distributed power allocation for sink-centric clusters in multiple sink wireless sensor networks. Sensors, **10**, 2003–2026 (2010). doi:10.3390/s100302003
17. Y-h Zhu, W-d Wua, J Pan, Y-p Tang, An energy-efficient data gathering algorithm to prolong life-time of wireless sensor networks. Comput Commun. **33**, 639–647 (2010). doi:10.1016/j.comcom.2009.11.008
18. S Soro, WB Heinzelman, Cluster head election techniques for coverage preservation in wireless sensor networks. Ad Hoc Netw. **7**, 955–972 (2009). doi:10.1016/j.adhoc.2008.08.006
19. A Sharaf, J Beaver, A Labrinidis, P Chrysanthis, Balancing energy efficiency and quality of aggre-gate data in sensor networks. VLDB J. **13**(4):384–403 (2004). doi:10.1007/s00778-004-0138-0
20. NA Vasanthi, S Annadurai, Energy efficient sleep schedule for achieving minimum latency in query based sensor networks, in *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, Workshops*. **2**, 214–219 (June 5-7, 2006)

doi:10.1186/1687-1499-2011-73
**Cite this article as:** Mo *et al.*: **Clustering-based routing for top-k querying in wireless sensor networks.** *EURASIP Journal on Wireless Communications and Networking* 2011 **2011**:73.