Routing in Multi-Sink Sensor Networks Based on Gravitational Field

Jinbao Li, Shouling Ji, Hu Jin and Qianqian Ren
School of Computer Science and Technology
Heilongjiang University, Harbin, China
{jbli, slji, hujin}@hlju.edu.cn

Abstract

The process of data forwarding in sensor networks is analogy to electric charge moving in electrostatic field. By this analogy, a method which abstracting a sensor networks to a Gravitational Field is proposed in this paper. In this gravitational field, sink node has gravitational to the data and data can flow to sink under this gravitational. Based on this gravitational field, a routing method which applies well in Multi-Sink sensor networks is proposed in this paper. This method has a lower time and space complexity, and it can adapt to the variety of the networks size dynamically. Theoretic analysis and simulation results indicate that: the routing method this paper proposed can decrease the energy consuming of data transmission effectively, reduce the data packet discard rate, uniform the networks’ loads, and prolong the lifecycle of the networks.

1. Introduction

As the development of communications technology, embedding technology and microelectronics technology, sensor networks have been studied extensively in recent years. This networks have many applications including military, environment monitoring, medical treatment, building status monitoring and intelligence furniture and other many fields. A typical sensor networks is composed of sensor nodes and sink nodes. Generally, sensors’ computing capability, communicating ability and storage memory are weaker, sinks have more resources, they have enough processing power, storage space and capability of communicating. When a sensor node senses an event, it should forwarding the data to the sink through other mid-sensors. Sinks could communicate with the satellite, and by this way, users can interact with the networks.

In the present, some researchers map the problems in sensor networks to the classical physical problems, and solving these problems by the mathematics-physics methods. S.Toumpis and other researchers studied the optimal design and operation of massively dense wireless networks. They abstract the problem to the distributing problem of electric charges in the electrostatics. The networks optimal deployment conditions can be obtained through the charges distributing characteristic in the electrostatic field[1]. In [2,3], they studied this problem farther more, which equipped with a general physical layer.

M.Kalantari and M.Shayman studied the routing problem in Ad Hoc networks by analogy to electrostatic theory. In this abstracted electrostatic field, the authors get a series of Partial Differential Equations (PDEs) based on the properties of electrostatic field. By solving these PDEs, they can get the corresponding route of every sensor[4]. The same authors continued their work in [5]. In this paper, they studied the optimal design problem of Multi-Sink Sensor Networks also by analogy to electrostatic theory. In [6], they expanded their work to Multi-Commodity sensor networks; they proposed a routing method based on PDEs in this paper.

But, the algorithms based on PDEs have many disadvantages. Generally, we can get the PDEs' numerical results only in some special conditions, and the cost to figure out the PDE is too high. So, in the actual sensor networks, it is almost impossible to solve Partial Differential Equations by the present nodes.

A method to deploy multiple sinks in the sensor networks based on an electrostatic model was proposed in [7]. This method can accommodate the electric charges’ type and amount of the sensors dynamically,
by this way, it can build an energy-efficient networks. This method considers the residual energy mostly; when the mobile sinks change its place. But it ignores the environments of the networks running. In [8], a decentralized service discovery mechanism for ad hoc networks, which uses the field theoretic approach, was proposed. This method uses the positive and negative charge to model the service and service request separately; the service request can find its route to the destination through the potential values of its neighbors.

In [9], a simple Magnetic Diffusion (MD) mechanism based on magnetic filed was proposed. In this method, the sink is modeled by magnet and the data is modeled by metallic nails, and data can be forwarded to the sink according to the magnetic charge of every node. MD method can perform well in timely delivery data, achieve high data reliability and energy efficiently, but MD could not use the networks resources uniformly. A routing method by analogy to Geometrical Optics was proposed in [10, 11]. This method abstracts the process of data transmission in networks as the light spreading in some media with different refractive index.

After considering the energy consuming, load balancing and the lifecycle and other problems in the networks synthetically, we proposed a routing method in Multi-Sink sensor networks based on field theory in this paper. This method abstracts the sensor networks to a gravitational field generated by sinks, and it uses gravitational field intensity to denote the communication attributes of every sensor. And this method can choose the next-hop for all the sensors based on the gravitational field intensity of its neighbors. Theoretic analysis and simulation results indicate that compared with other similar algorithms, the routing method this paper proposed has a lower computational complexity, a lesser energy consuming and can improve the routing efficiency significantly. And the experiment results proved that it can adapt the Multi-Sink sensor networks well.

The remainder of this paper is organized as follows: in Section 2 we proposed the routing method in Multi-Sink sensor networks based on gravitational field, and we analyze its performance in this section too. In Section 3, we give some experiments, and compare the results with some other methods. And we conclude the paper in Section 4.

2. Routing mechanism based on gravitational field

According to the actual applications of sensor networks, we make the following assumptions: (1) the upper limit of networks load is known in some region; (2) the initial energy of a node is known, and the residual energy is known at any moment; (3) sink can communicate with the satellite, users can access the data via satellite.

Before the formal discussions, we make the following definitions first.

**Definition 1: Monitoring Area of Sensor.** The monitoring area of a sensor is a round region, and its radius is smaller than the sensor’s communications radius.

**Definition 2: Load of Sensor.** The total amount of data generated in the monitoring area of a sensor is the load of sensor.

**Definition 3: Monitoring Area of Sink.** If a sensor’s data is transmitted to the sink, then the monitoring area of this sensor is belonged to the monitoring area of this sink. The monitoring area of a sink is equal to the total monitoring area of all sensors, which transmit their data to this sink.

**Definition 4: Load of Sink.** The total amount of data generated in the monitoring area of a sink is the load of sink.

2.1. Definition of gravitational field

We know from electromagnetism, electric charge can generate electrostatic filed, and it has strength effect to other charges in this filed. Suppose that an electrostatic field is generated by a negative charge, and then it has an attraction to a positive charge in this filed. The positive charge moves toward to the negative charge under this attraction.

Consider the data flows form the sensor to the sink by multi-hop way in the sensor networks; it is analogy to the charges interaction and moving in the electrostatic field. By this analogy, we make the following abstracting:

The data that monitored by sensors is modeled by positive charge with appropriate magnitude; the sinks are modeled by negative charge with appropriate magnitude, and sinks have attraction to the data. In this way, the networks is abstracted to a field generated by sinks. In this field, sensors may generate data randomly, and sinks have attraction to these data, and the data can flow to the sinks under this attraction.

By the analogy to charges moving in electrostatic field, we give the definition of Gravitational Field, Attraction of Data, and Gravitational Field Intensity first time in the following.

**Definition 5: Gravitational Field (GF).** Consider a single-sink sensor networks deployed in area A, the coordinate of the sink is $(x_s,y_s)$. Then, if a sensor in the networks monitors some data, the data transmission can abstract as: sink has an attraction to the data, and
the data will flow to the sink under this attraction. In this way, the networks deployed in area \( A \) is abstracted as a Gravitational Field generated by the sink in \((x_s, y_s)\).

**Definition 6: Attraction of Sink (AoS).** After the above abstracting process, for a sensor in the networks which coordinate is \((x, y)\), define the Attraction of Sink to the data sensed by this sensor as follows:

\[
\tilde{F}(x, y) = k(x, y) \frac{Q - q(x, y)}{r^2} \hat{r}
\]

in which \( \hat{r} \) is a unit vector, and it denotes the direction of the attraction; \( Q \) is the load of sink; \( q(x, y) \) is the unit load, namely, the data amount generated in per unit of area per unit of time. \( r \) is the logistic distance between the sensor in \((x, y)\) and the sink. The definition of \( r \) is:

\[
r = d_{s,x,y}(x, y) \cdot w(x, y)
\]

in which \( d_{s,x,y}(x, y) \) is the physical distance between \((x, y)\) and sink. \( w(x, y) \) is the environment function in location \((x, y)\), which reflects the influences to the communications brought by landform, physiognomy, barrier, the weather conditions and other environment factors. Consider the environmental characteristics, can illuminate the communications capacity of a certain place better.

\( k(x, y) \) in Equation(1) is the energy function, its definition as follow:

\[
k(x, y) = a \frac{E_f(x, y)}{E_f(x, y)}
\]

in which \( a \) is energy function factor; \( E_f(x, y) \) is the actual energy of the sensor in \((x, y)\), according to assumptions, it is known at any moment; \( E_f(x, y) \) is the energy of the sensor in \((x, y)\) under ideal conditions.

In the electric field theory, in order to describe the properties in a certain point, they use the concept of electric intensity. Analogously, we use Gravitational Field Intensity to describe the properties in a certain point in the Gravitational Field.

**Definition 7: Gravitational Field Intensity (GFI).** The Gravitational Field Intensity of a sensor in \((x, y)\) is the ratio of Attraction of Sink to the data in this sensor and the unit load. It can be written as:

\[
\tilde{C}(x, y) = \tilde{F}(x, y) \frac{Q}{q(x, y)} \hat{r}
\]

in which \( \tilde{F}(x, y) \) is the Attraction of Sink to the data in this sensor; \( q(x, y) \) is the unit load. From equation (4), we know: the GFI can reflect the influences to the communications made by the physical distance between sensor and sink, the residual energy at some certain moment and the environmental characteristics and other factors.

The definitions above can be extended to Multi-Sink case straightforwardly. Consider a Multi-Sink sensor networks deployed in area \( A \) contains \( M \) sinks, and these \( M \) sinks generate a Gravitational Field in Area \( A \) together. For a sensor in this network which coordinate is \((x, y)\), the Attraction of Sink to the data in this sensor is:

\[
\tilde{F}(x, y) = \tilde{F}_1(x, y) + \tilde{F}_2(x, y) + \cdots + \tilde{F}_M(x, y) = \sum_{i=1}^{M} \tilde{F}_i(x, y)
\]

in which \( \tilde{F}_i(x, y) \) is the attraction generated by the \( i \)th sink. Then, the GFI of the sensor in \((x, y)\) is:

\[
\tilde{C}(x, y) = \tilde{C}_1(x, y) + \tilde{C}_2(x, y) + \cdots + \tilde{C}_M(x, y) = \sum_{i=1}^{M} k(x, y) \frac{Q_i}{r_i^2} \hat{r}_i
\]

in which \( Q_i \) is the load of the \( i \)th sink; \( r_i \) is the logistic distance between \((x, y)\) and the \( i \)th sink.

2.2. Routing mechanism based on gravitational field

2.2.1. Solving of GFI. Form the definition of GFI in section 2.1 we know: when compute the GFI of every sensor, we don’t consider its neighbors, but only consider the residual energy, the environmental characteristics, Etc. of the current node. So, we can figure out the GFI of every sensor in the networks by a distributed parallel algorithm.

Consider a Multi-Sink sensor networks contains \( m \) sinks and \( n \) sensors. In order to get the GFI of the \( n \) sensors, we only need these \( n \) sensors calculate equation (6) respectively. We propose the following Gravitational Field Intensity Algorithm (GFIA) to calculate the GFI of every node. The basic thinking of GFIA is: i) for every sensor, compute the components of GFI separately generated by every sink; ii) compute the vector adding result of these GFI components, and the eventually result is the sensor’s GFI. In the following GFIA, Gralten is the data structure of GFI, and it composed by the magnitude of GFI, the direction of GFI, Etc. NodelInfo is the data structure of node information. It contains the node’s coordinate, residual energy and the information of surroundings.

**Algorithm 1: Gravitational Field Intensity Algorithm (GFIA)**

1: Gralten \( G_i = \) Null, \( g \); 2: NodelInfo sensor\_i, sink\_m; 3: for sensor \( i \) 3.1: for \( j=0; j<m; j++ \) 3.1.1: according to sink\_j, sensor\_i and eq.(4), compute the GFI component generated by sink\_j, and store it into \( g \); 3.1.2: compute the vector adding result of \( g \) and \( G_i \), and store it into \( G_i \); 4:end;

Suppose compute eq.(4) in 3.1.1 step in the above GFIA will cost \( t_1 \) in time, and the 3.1.2 step in GFIA
will cost $t_2$ in time, the number of sinks in the networks is $m$, then the complexity of computing time to calculate the GFI of every sensor is $O(m^* (t_1 + t_2))$. Suppose the data structure space of NodelInfo type is $s_1$, and the data structure space of Graph type is $s_2$, then the complexity of computing space of GFIA is $O((m+1)* s_1+2s_2)$. Generally, the number of sinks in a sensor networks is very small, hence the computational complexity of GFIA is between the constant level and polynomial level. Besides this, the computational complexity of GFIA is only related to the number of sinks, but unrelated to the number of sensors, therefore GFIA can adapt to changes of the networks size dynamically.

2.2.2. Routing mechanism. When a sensor (we denote this sensor by $S$) has some data (the data may be sensed by the sensor itself or just received from other nodes and forward it to the next hop) to transmit, the first thing is to determine the next-hop. In the abstract gravitational field, GFI describes a sensor’s residual energy, surroundings, the distance to every sink and other multi-parameters comprehensively. If a node has a bigger GFI in numerical value, it represents that this node has more residual energy, the surroundings has less interferences to the communications, and it has a closer distance to the sink. Hence, when $S$ chooses its next-hop, it chooses the neighbor node which has the biggest GFI in numerical value. In this way, not only the energy can be used uniformly, but also the reliability of transmission can be improved. Here, we call the neighbor node of $S$ with the biggest GFI in numerical value the optimal next-hop, and we denote it by $S_{\text{next}}$.

When determine the route corresponding to $S$, we choose $S$’s optimal next-hop $S_{\text{next}}$ to be the next station of data transmission. If $S_{\text{next}}$ it the sink, we use the same strategy to choose the optimal next-hop for $S_{\text{next}}$, until the next-hop is sink.

According to the definition of GFI, we know that the magnitude of GFI is inversely proportional to the square of distance between the sensor and sink. Hence, the contour maps of GFI are a series of circles with different radiuses, and the centre of these circles is the sink. If a sensor’s location is in the vicinity of a circle with a smaller radius, i.e. it is closer to the sink, and then its GFI is bigger in numerical value. Conversely, its GFI is smaller. Furthermore, the GFI has a direction, and its direction points at the sink, i.e. the direction has a bigger GFI in numerical value, consider the definition of the optimal next-hop, the location of the optimal next-hop is closely related to the direction of GFI. Based on this fact, we proposed the Orientation Based Next Hop (OBNH) Algorithm to choose the optimal next-hop for a sensor in the sensor networks.

![Figure 1. (Adjacency)Azimuth angle and (adjacency)azimuth region](image)

Before the description of OBNH, we give the following definitions used in the algorithm:

**Definition 8: Azimuth Angle.** As showed in Fig.1, suppose the GFI of the sensor in $(x,y)$ is $a(x,y)$ (bold-face denote vector). Define the angle which magnitude is $A$, and takes $(x,y)$ to be its vertex, and takes $a(x,y)$ to be its angle bisector and its opening direction along the direction of GFI as the Azimuth Angle of the sensor in $(x,y)$. In Fig.1, angle NOM is the Azimuth Angle corresponding to the sensor in $(x,y)$.

**Definition 9: Azimuth Region.** In Fig.1, the dotted arc is part of the circle which centre is $(x,y)$ and the radius is the communications distance of the sensor in $(x,y)$. Define the region surrounded by $OM\ ,\ ON$ and arc $MN$ as the Azimuth Region of the sensor in $(x,y)$, and we denote it by $A\text{-area}$. When the OBNH choosing the optimal next-hop for the sensor in $(x,y)$, because of the location of the optimal next-hop is closely related to the direction of GFI, therefore OBNH will search the sensor’s appropriate next-hop in $A\text{-area}$ firstly.

**Definition 10: Adjacency Azimuth Angle.** If OBNH couldn’t find the sensor’s appropriate next-hop in $A\text{-area}$, it should expand the search scope. Taking both sides of the Azimuth Angle circumvolve outward with an angle which magnitude is $B/2$, define these two new angles with the magnitude $B/2$ as the Adjacency Azimuth Angles. In Fig.1, the Adjacency Azimuth Angles of the sensor in $(x,y)$ are angle NOP and angle MOQ.

**Definition 11: Adjacency Azimuth Region.** The new additional search region corresponding to the Adjacency Azimuth Angle is defined as Adjacency Azimuth Region. In Fig.1, we define the region surrounded by dotted arc $NP$, line $ON\ ,\ OP$ and the region surrounded by line $OM\ ,\ OQ$ and dotted arc $MQ$ as the Adjacency Azimuth Region of the sensor in $(x,y)$. We denote it by $B\text{-area}$.

The four above definitions relate to the localization technology in the sensor networks. But these definitions only to narrow the search scope when OBNH chooses the next-hop for a sensor. Therefore,
here we don’t need the precise position information of the nodes. So, we can solve this problem by some localization algorithm or GPS.

We give the OBNH algorithm in the following. In OBNH, $S(x,y)$ represents the sensor in $(x,y)$; $S_{next\_hop}$ represents the next-hop obtained by OBNH.

<table>
<thead>
<tr>
<th>Algorithm 2: Orientation Based Next Hop Algorithm (OBNH)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: set the communication radius $dis$;</td>
</tr>
<tr>
<td>2: set of neighbor nodes $S_{neighbor}$ = NULL;</td>
</tr>
<tr>
<td>3: set Azimuth Angle $A$&gt;0, Adjacency Azimuth Angle $B$=0;</td>
</tr>
<tr>
<td>4: compute the Azimuth Region $A$-area corresponding to $S(x,y)$;</td>
</tr>
<tr>
<td>5: while $(A+B&lt;2\pi)$</td>
</tr>
<tr>
<td>5.1: compute the Adjacency Azimuth Region $B$-area corresponding to $S(x,y)$;</td>
</tr>
<tr>
<td>5.2: initialize $S_{neighbor}$ according to $A$-area and $B$-area;</td>
</tr>
<tr>
<td>5.3: set $S_{next_hop}$=NULL, the GFI of $S_{next_hop}$ $</td>
</tr>
<tr>
<td>5.4: while $(S_{neighbor}!= NULL)$</td>
</tr>
<tr>
<td>5.4.1: take a node $S(x^<em>,y^</em>)$ form $S_{neighbor}$;</td>
</tr>
<tr>
<td>5.4.2: if $(S(x^<em>,y^</em>)$ is a sink )</td>
</tr>
<tr>
<td>5.4.2.1: $S_{next_hop}= S(x^<em>,y^</em>)$; break;</td>
</tr>
<tr>
<td>5.4.3: else if the GFI of $(S(x^<em>,y^</em>)</td>
</tr>
<tr>
<td>5.4.3.1: $</td>
</tr>
<tr>
<td>5.4.4: $S_{neighbor}=S_{neighbor}- {S(x^<em>,y^</em>)}$;</td>
</tr>
<tr>
<td>5.5: if $(S_{next_hop} != NULL)$</td>
</tr>
<tr>
<td>5.5.1: return $S_{next_hop}$;</td>
</tr>
<tr>
<td>5.6: else</td>
</tr>
<tr>
<td>5.6.1: $A+=B$, reset Adjacency Azimuth Angle $B&gt;0$;</td>
</tr>
</tbody>
</table>

At the beginning of the above algorithm, OBNH sets the Adjacency Azimuth Angle to be 0, and searches the next-hop in the Azimuth Region $A$-area. Take the sensor which coordinate is $(x,y)$ in Fig.1 as an example, OBNH search the next-hop in the neighbor nodes set $\{S1,S2\}$ first. If OBNH can find the appropriate next-hop in this set, it will end after returning the appropriate next-hop. Else, OBNH will search the next-hop in the Adjacency Azimuth Region $B$-area. In Fig.1 the neighbor nodes set corresponding to $B$-area is $\{S3,S4,S5,S6\}$. If the OBNH still can’t find the appropriate next-hop, it will reset the Adjacency Azimuth Angle and the Adjacency Azimuth Region again until find the appropriate next-hop.

### 2.3. Performance analysis

To the GFI proposed in this paper, by the analysis in section 2.2.1, the complexity of computing time and space are all under the constant level and polynomial level, and it only rely on the number of sinks in the networks. Therefore, the routing method based on gravitational field proposed in this paper can reduce the computational complexity both in time and space significantly. When the networks size is increasing, the cost of GFI has no obvious increasing because its computational complexity is only rely on the number of sinks in the networks.

### 3. Simulation results and analysis

Suppose there is a sensor networks deployed in an area. Sensors uniformly distributed and sinks randomly distributed in the networks. For the simplicity of the experiments, here we consider a networks deployed in a 19×19 grid. There are 4 sinks and 257 sensors in the networks, and the coordinates corresponding to each sink are: $(x_1,y_1)=(9,9)$, $(x_2,y_2)=(15,15)$, $(x_3,y_3)=(13,5)$, $(x_4,y_4)=(5,15)$. Suppose the upper limit of networks load is 100, and the load distributes uniformly in the networks, and the initial load of every sink is the same. Suppose the surroundings of nodes in the networks are same, i.e. $w(x,y)$ is a constant. Furthermore, we set energy function $k(x,y)=1$ in our experiments.

We obtained the GFI of every sensor in the networks as shown in Fig.3 by using GFI proposed in section 2.2.1. The direction of GFI in every sensor is shown in Fig.4. By running OBNH, we can get the route corresponding to each sensor. And the weights of each sink are: $Q(x_1,y_1)=29.6399$, $Q(x_2,y_2)=21.8836$, $Q(x_3,y_3)=25.2078$, $Q(x_4,y_4)=23.2687$.

In order to study the method proposed in this paper further, we study the performance of this method when it works in different sensor networks with different sinks. And we will compare its performance with Directed Diffusion (DD) [12] Algorithm and the Flooding Algorithm. Suppose there are 19×19 nodes in the networks, and the deployment of the networks is similar to the settings of the above experiment but with different sinks in numbers and locations. The surroundings of nodes in the networks are same. The initial energy of a sensor is 1000 units, and the initial energy of a sink is 10000 units. Sensors are uniformly distributed in the networks and sinks are randomly distributed in the networks.

![Figure 3. GFI of every sensor in the networks](image-url)
i.e., the directly adjacent neighbors. If an event has two or more nearest nodes, then it can be sensed by one of these nearest nodes in the same probability. Of course, if the event’s nearest nodes contain a sink; it will be sensed by the sink preferentially. Monitoring an event successfully will cost 1 unit energy, and transmitting or receiving a data packet will cost 1 unit energy separately too. Suppose a sensor can only communicate with its directly adjacent neighbors, its biggest communication radius is $\sqrt{2}$, i.e. the directly adjacent neighbors of a sensor can be 3, 5 or 8. In order to compare different networks with different number of sinks and working in different routing methods, here we consider the scenarios of networks has 1-sink to 10-sink severally. In which, $i$-sink denotes a sensor networks with $i$ sinks.

Before the study, we make the following definitions first:

**Definition 12: Life Cycle of Sensor Networks.** The Life Cycle of Sensor Networks begins with the deployment of the networks and will end if all the sinks in the networks can not receive any data packets.

**Definition 13: Packet Discarding Rate.** If a data packet can’t be transmitted to the sink, and then it will be discarded. The ratio of data packets discarded and the total amounts of data packets is defined as Packet Discarding Rate.

When the above $i$-sink ($1 \leq i \leq 10$) networks working in the routing method (we denote it by GFIA) proposed in this paper, DD algorithm and Flooding algorithm separately, the biggest total load of different networks shown in Fig.5. In Fig.5, $x$-axis denotes different networks with different number of sinks; $y$-axis denotes the average number of packets sensed by per sink. Form Fig.6 we know: if DD has a long routing updating cycle, the average number of packets sensed by per sink is more. If the networks has less sinks, DD is better than GFIA. But when the networks has more sinks, because GFIA can use the networks resources more uniformly, it is better than DD. And the average number of packets monitored by every sink can be maintained at around 4400. Certainly, when the number of sinks achieves a certain magnitude, it will lose its meaning to add sinks more.

Fig.7 shows the average hops of the packets form being generated to be transmitted to a sink. In which, $x$-axis denotes different networks with different number of sinks; $y$-axis denotes the average number of packets sensed by per sink. We can see from Fig.7, the average hops of data packets decrease as the increase of sinks. The average packets hops of GFIA is more than DD, the reason can be obtained from Fig.7. When GFIA chooses routes for some sensors, it chooses a longer path. This choice can not only avoid congestions and
reduce conflicts but also use the energy more uniformly. The average energy cost of every data packet is shown in Fig.8. In Fig.8, x-axis denotes different networks with different number of sinks; y-axis denotes the average energy cost of every packet. As the increase of sinks, the average hops of every packet is decrease, hence the average energy cost is depressed too. Although the average hops of packets in GFIA is more, its average energy cost is lower than DD. This is because the Packet Discarding Rate is higher; it wastes a lot of energy. And DD updates its routing periodically; it also consumes part of the additional energy. We give the Packet Discarding Rate of 4-sink networks in Fig.9.

![Figure 7. Average hops of data packets](image)

![Figure 8. Average energy cost of every data packet](image)

![Figure 9. Packet discarding rate of 4-sink networks](image)

4. Conclusions and future work

We proposed a routing method (GFIA) in Multi-Sink sensor networks based on field theory in this paper. This method abstracts the sensor networks to a Gravitational Field generated by sinks. In this abstract field, sink had attraction to the data, and data can flow to the sink under this attraction. When study the problem above, this method considered the influences of the distance between sensor and sink, the residual energy of sensor, the environmental characteristics and other factors. Compared with the routing method based on PDEs, DD and Flooding: i) the computational complexity of GFIA is lower; ii) GFIA can adapt the changes of the networks size dynamically; iii) GFIA can solve the problems of energy-efficiency and load balancing effect is more better and it can prolong the lifecycle of the networks by increasing the total tasks that can by accomplished by the networks.
balancing well. And GFIA can improve the performances of Multi-Sink sensor networks significantly.

The future work includes solving some additional issues in GFIA. For example: the surroundings of sensors in the networks are impossible to be all the same. How to choose environment function need further study.

5. References


