A DYNAMIC BUFFER ZONE BASED ALGORITHM FOR SINGLE MOBILE SINK IN WIRELESS SENSOR NETWORKS

Zhi Huang[1], Sanyang Liu[2] and Xiaogang Qi[3]

[1]School of Computer Science & Technology, Xidian University, Xi'an, China
huangzhi2003@126.com
[2] Department of Applied Mathematics, Xidian University, Xi'an, China
liusanyang@126.com
[3] Department of Applied Mathematics, Xidian University, Xi'an, China
qixiaogang@gmail.com

ABSTRACT

Limit energy is a severe bottleneck of wireless sensor networks (WSNs), and limits network lifetime of WSNs. To extend network lifetime, buffer zone has been proposed. Sensors send their data packets to buffer zone. The sensors in buffer zone buffer the data packets. And then the sink visits the buffer zone to collect the data packets. This leads to that the loads of the sensors in buffer zone are too high and the sensors die quickly. To further extend network lifetime, an algorithm based on dynamic buffer zone has been proposed in this paper. The algorithm divides the whole network area into some areas, and lets all areas act as the buffer zone in turn. And the reasonable times each zone acts as the buffer zone are computed with linear programming. The simulation results have shown that our proposed algorithm notably extends network lifetime.

KEYWORDS

Wireless Sensor Networks, Mobile Sink, Buffer Zone, Linear Programming

1. INTRODUCTION

Wireless sensor networks is an important potential technology and can be applied to many applications, such as industry, agriculture, military, space exploration, and so on [1-4]. Energy resource is a severe bottleneck of wireless sensor networks because sensors are usually powered by light weight batteries and these batteries usually can't be charged or replaced [3]. Therefore, how to effectively utilize energy resource of sensors is an important issue of wireless sensor networks. One important approach is sink mobility. That is, sink moves in the network area in order to shift the burden from sensors to the sink [5].

To further extend network lifetime, another approach, deploying both mobile sink and buffer zone in network area, has been proposed [6]. A part of network area act as buffer zone. All sensors send their data packets to the buffer zone. Then the sensors in the buffer zone buffer these data packets, and directly send these data packets to the sink when the sink moves into their communication range. The sink moves back and forth in the buffer zone to collect data packets. The approach extend network lifetime notably. However, sensors in the buffer zone forward too many data packets, consume much energy and die quickly. This limits network lifetime.

DOI : 10.5121/ijcsea.2012.2502

In this paper, to further extend network lifetime, we have proposed a dynamic buffer zone based algorithm for single mobile sink in wireless sensor networks. The algorithm divides the network area into some sub areas. Each sub area acts as the buffer zone in turn. And the time each sub area acts as the buffer zone are reasonably computed with linear programming.

The remainder of this paper is organized as follows:

2. RELATED WORK

Sink mobility has been widely proposed. And many algorithms for sink mobility have been proposed.

Data Mules [7] was proposed aiming at sparse sensor network. Mobile entities, such as people, animals, and cars, are called Data Mules. An entity moves in network area, collect and buffer data from a sensor when the distance between them is short enough. An entity sends data to an Access Point when they are near. The algorithm saves energy. However, it leads to tremendous latency.

Ref. [8] assumes that sensors are deployed in a circle area. It is proved that moving along the periphery of the circle network area is optimum. Ref. [6] has studied the same network model, and proposed an algorithm with buffer zone, which is called MADG (movement-assisted data gathering). The algorithm deploys both mobile sink and buffer zone in network area. A ring area acts as the buffer zone. The sensors in the buffer zone buffer all sensors' data. The sink moves back and forth in the buffer zone, and collect data when it is near to a sensor, as shown in figure 1. The algorithm extend network lifetime notably. However, the sensors in buffer zone consume much energy and die quickly. In this paper, we have improved the algorithm and proposed a new algorithm.

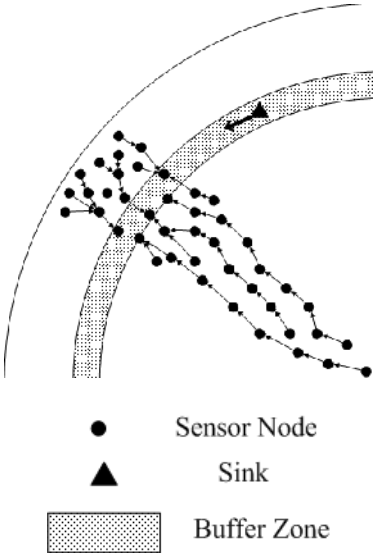

Figure 1. MADG (Only a part of network is drawn to simplify the figure)

3. NETWORK MODEL

N sensors are randomly distributed in a circle area with the radius R and the centre O. The communication radius of sensors is r(r<<R). The density of sensors is high. Therefore, the

network has strong connectivity [6]. Each sensor knows its own position and the position of the centre, as well as the radius R. The sink is mobile, can move in the network area, and has limitless power.

4. A DYNAMIC BUFFER ZONE BASED ALGORITHM FOR SINGLE MOBILE SINK

In MADG, all sensors' data packets are sent to and buffered by the sensors in the buffer zone. Then a sensor in the buffer zone directly sends data to the sink when the distance between the sensors and the sink is short enough. The topology in Figure 1 can be abstracted as Figure 2. As seen in Figure 2, all sensors' data packets are forwarded by the sensors in the buffer zone. Therefore, the sensors in the buffer zone forwards many data packets, consume much energy, and die quickly. This limits network lifetime. To further extend network lifetime, we have proposed the approach which partitions the whole network area into some sub areas, let each sub area act as the buffer zone. The time each sub area act as the buffer zone is computed with the linear programming in which the target function is maximizing network lifetime. Therefore, the balance of energy consumption can be obtained and network lifetime can be extended.

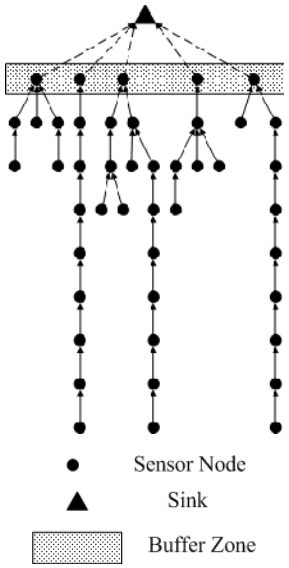

Figure 2. Abstraction of the topology in Figure 1

The following notation is used in describing our algorithm:

r	communication radius of sensors
e	initial energy of sensors
s	the number of rings
d	the distance between two adjacent rings
n	a sensor in network
N	the set of all sensors in network
k(n)	the number of the ring in which sensor n is

d(n)	the distance from the centre to sensor n
nb(n)	the set of all neighbor of sensor n
nb_1(n)	the set of the sensors which belong to and is in the ring k(n)-1
nb_2(n)	the set of the sensors which is in ring k(n) and d(x)<d(n)
nb_3(n)	the set of the sensors which is in ring k(n) and d(x)>d(n)
nb_4(n)	the set of the sensors which belong to and is in the ring k(n)+1
u(n)	the next hop of sensor n when data is forwarded towards the centre
d(n)	the next hop of sensor n when data is forwarded towards the periphery
l_{ij}	the maximum load (the energy consumed in a unit time) of the ring j($1 \leq j \leq s$) when the ring i($1 \leq i \leq s$) acts as the buffer zone
t_i	the time the ring i acts as the buffer zone

4.1. Network Partitioning

The whole circle network area are evenly partitioned into s concentric rings, s=R/d(d<r), where the centre of the rings are the centre of the circle network area. And the distance between the rings is d, as shown in Figure 3. All rings are outward numbered as ring 1, ring 2, ..., ring s. And the time each ring acts as the buffer zone is $t_i(1 \leq i \leq s)$.

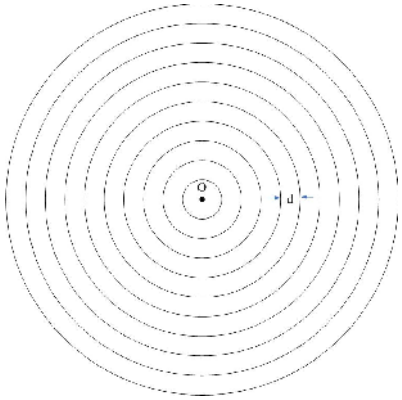

Figure 3. Partitioning of network

4.2. Routing

Once a network is deployed and starts, for sensor n, two sensors, d(n) and u(n) are found. When data is transferred towards the centre of the circle network area, n transfers its data to d(n). And when data is transferred towards the border of the circle network area, n transfers its data to u(n). The approach to find d(n) and u(n) are described as follows.

The set of all neighbour sensors of n, nb(n), is divided into four subsets, nb_1(n), nb_2(n), nb_3(n) and nb_4(n), as shown in figure 4. In nb_1(n), the senor with the shortest distance to the centre is found, and its ID is assigned to u(n). If nb_1(n) is empty, then the senor with the shortest distance to the centre is found in nb_2(n), and its ID is assigned to u(n). In nb_4(n), the senor with the longest distance to the centre is found, and its ID is assigned to d(n). If nb_4(n) is empty, then the senor with the shortest distance to the centre is found in nb_3(n), and its ID is assigned to d(n).

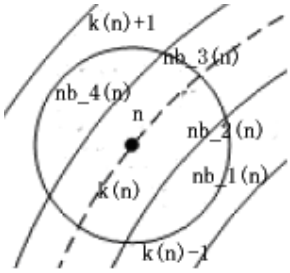

Figure 4. Partitioning of network

The algorithm to find d(n) and u(n) are described as follows.

```
/* The algorithm that is used by a sensor node to find its U(n) and D(n) */
1   for each n∈N do
2           find nb(n)
3           find nb_1(n),nb_2(n), nb_3(n), nb_4(n)
4           if n is not in ring 1
5                   if nb_1(n) is not a empty set
6                           find the node x which is in nb_1(n) and has the min d(x) in nb_1(n)
7                           u(n)=x
8                   end if
9                   else
10                          find the node x which is in nb_2(n) and has the min d(x) in nb_2(n)
11                          u(n)=x
12          end if
13          if n is not in ring s
14                  if nb_4(n) is not a empty set
15                          find the node x which is in nb_1(n) and has the max d(x) in nb_4(n)
16                          u(n)=x
17                  end if
18                  else
19                          find the node x which is in nb_3(n) and has the max d(x) in nb_3(n)
20                          u(n)=x
21          end if
22  end for
```

4.3. Times Each Zone Acts as the Buffer Zone

The times each zone acts as the buffer zone must be reasonably computed to extend network lifetime. And linear programming is adopted to calculate the times, as follows.

$$\begin{cases} MaxZ = \sum_{i=1}^{s} t_i \\ S.t \\ t_1 * l_{11} + ... + t_i * l_{i1} \le e \\ ... \\ t_1 * l_{1j} + ... + t_i * l_{ij} \le e \\ t_1,...t_i > 0 \end{cases} \qquad (1)$$

The reasonable times each zone acts as the buffer zone can be obtained with working out the linear programming.

The maximum load of the ring j when the ring i acts as the buffer zone, l_{ij}, is difficult to be calculated because it relates to specific network topologies and network topologies are randomly generated. Actually, l_{ij} is determined by the following six factors: R (the radius of the network area), r (communication radius of sensors), |N| (the number of sensors), D (network partitioning) as well as the values of i and j. The first four factors can be curtain before the network operation. Therefore, we can repeatedly simulates on computers, and obtain the average value as the expectation of l_{ij}.

4.4. Implementation of the Algorithm

The algorithm is implemented as the following steps:

(1) Before a network is deployed, since we know the values of R, r, |N| and D of the network to be deployed, computer simulations can be made with these values. And then the values of all l_{ij} ($0 \le i,j \le s$) can be obtained. Thereby the values of all t_{ij} ($0 \le i,j \le s$) can bee calculated with equation (1).

(2) After the network is deployed and starts working, ring x (the initial value of x is 1) acts as the buffer zone for time t_x. The sensors in the rings x+1~s (when x<s) transfer their data packets to their u(n), the sensors in the rings 1~x-1 (when x>1) transfer their data packets to their d(n). The sensors in the ring x receive data packets from other sensors. When the sink moves into their communication ranges, the sensors in the ring x transfer the data packets they receive and their own data packets to the sink.

(3) After the ring x acts as the buffer zone for time t_x, the sink, broadcasts a message to all sensors to inform that the ring x+1 starts acting as the buffer zone (lets x=x+1), and starts moving in ring x+1 (when x<s). After receiving the message, the ring x+1 (when x<s) acts as the buffer zone with the method of Steps (2).

(4) Steps (2) and (3) are repeated until the network dies. If the network does not die after the ring s acts as the buffer zone, the ring s continues acting as the buffer zone until the network dies.

5. SIMULATION RESULTS

Physic layer and MAC layer are ignored in simulations because we mainly focus on the upper data gathering algorithm. To simplify simulations, we assume that each sensor generates a data packet in one unit time, and one unit energy is consumed to send a data packet. We have assumed that a network area with the radius R=100m. 3000 sensors with the communication radius r=12m are randomly deployed in the network. The initial energy of sensors is 5000 units. Each sensor generates 1 data packet per time unit. 1 unit energy is consumed to transfer a data packet.

Three algorithms are compared in the simulations. The first one is static sink (short for “SS”), the sink always stays at the centre of the network area during network operation. The second one is MADG and the third one is proposed DBDG.

In DBDG, we let s=10. That is, the network area is evenly partitioned into 10 concentric rings. The expectations of l_{ij} are obtained through simulations, as shown in table.

Table 1. The expectations of l_{ij} are obtained through simulations

Ring Ring	1	2	3	4	5	6	7	8	9	10
1	1251.7	295.7	174.2	115.5	82.0	53.9	35.3	18.5	10.6	3.8
2	1.0	296.7	174.2	115.5	82.0	53.9	35.3	18.5	10.6	3.8
3	1. 0	4.0	175.7	115.5	82.0	53.9	35.3	18.5	10.6	3.8
4	1. 0	4.0	8.4	117.1	82.0	53.9	35.3	18.5	10.6	3.8
5	1. 0	4.0	8.4	12.4	84.1	53.9	35.3	18.5	10.6	3.8
6	1. 0	4.0	8.4	12.4	20.1	56.1	35.3	18.5	10.6	3.8
7	1. 0	4.0	8.4	12.4	20.1	26.1	42.3	18.5	10.6	3.8
8	1. 0	4.0	8.4	12.4	20.1	26.1	33.0	43.7	10.6	3.8
9	1. 0	4.0	8.4	12.4	20.1	26.1	33.0	42.1	53.2	3.8
10	1. 0	4.0	8.4	12.4	20.1	26.1	33.0	42.1	50.8	65.9

And then the following linear programming is obtained.

$$
\begin{cases}
\text{Max } Z = t_1 + t_2 + t_3 + t_4 + t_5 + t_6 + t_7 + t_8 + t_9 + t_{10} \\
\text{S.t} \\
1251.7 t_1 + 1.0 t_2 + 1.0 t_3 + 1.0 t_4 + 1.0 t_5 + 1.0 t_6 + 1.0 t_7 + 1.0 t_8 + 1.0 t_9 + 1.0 t_{10} \le 5000 \\
295.7 t_1 + 296.7 t_2 + 4.0 t_3 + 4.0 t_4 + 4.0 t_5 + 4.0 t_6 + 4.0 t_7 + 4.0 t_8 + 4.0 t_9 + 4.0 t_{10} \le 5000 \\
174.2 t_1 + 174.2 t_2 + 175.7 t_3 + 8.4 t_4 + 8.4 t_5 + 8.4 t_6 + 8.4 t_7 + 8.4 t_8 + 8.4 t_9 + 8.4 t_{10} \le 5000 \\
115.5 t_1 + 115.5 t_2 + 115.5 t_3 + 117.1 t_4 + 12.4 t_5 + 12.4 t_6 + 12.4 t_7 + 12.4 t_8 + 12.4 t_9 + 12.4 t_{10} \le 5000 \\
82.0 t_1 + 82.0 t_2 + 82.0 t_3 + 82.0 t_4 + 84.1 t_5 + 20.1 t_6 + 20.1 t_7 + 20.1 t_8 + 20.1 t_9 + 20.1 t_{10} \le 5000 \\
53.9 t_1 + 53.9 t_2 + 53.9 t_3 + 53.9 t_4 + 53.9 t_5 + 56.1 t_6 + 26.1 t_7 + 26.1 t_8 + 26.1 t_9 + 26.1 t_{10} \le 5000 \\
35.3 t_1 + 35.3 t_2 + 35.3 t_3 + 35.3 t_4 + 35.3 t_5 + 35.3 t_6 + 42.3 t_7 + 33.0 t_8 + 33.0 t_9 + 33.0 t_{10} \le 5000 \\
18.5 t_1 + 18.5 t_2 + 18.5 t_3 + 18.5 t_4 + 18.5 t_5 + 18.5 t_6 + 18.5 t_7 + 43.7 t_8 + 42.1 t_9 + 42.1 t_{10} \le 5000 \\
10.6 t_1 + 10.6 t_2 + 10.6 t_3 + 10.6 t_4 + 10.6 t_5 + 10.6 t_6 + 10.6 t_7 + 10.6 t_8 + 53.2 t_9 + 50.8 t_{10} \le 5000 \\
3.8 t_1 + 3.8 t_2 + 3.8 t_3 + 3.8 t_4 + 3.8 t_5 + 3.8 t_6 + 3.8 t_7 + 3.8 t_8 + 3.8 t_9 + 65.9 \le 5000 \\
t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10} \ge 0
\end{cases}
\quad (2)
$$

After resolving the linear programming, the following result is obtained.

Z=146.31

t1=0.00	t2=0.00	t3=22.54	t4=8.2	t5=2.41
t6=8.62	t7=8.11	t8=11.40	t9=13.43	t10=71.56

That is to say, the network lifetime is 146.31 time units using proposed DBDG.

10 network topologies are randomly generated. SS, MADG and DBDG run in each topology. Two performance metrics, network lifetime, and total energy consumption are compared.

Network lifetime is defined as the time from the network starts operating to it dies. The simulation results are shown in Figure. As shown in the figure 5, DBDG notably improve network lifetime. Using DBDG, network lifetime is improved by 771.43% on average compared to SS, and 133.12% on average compared to MADG.

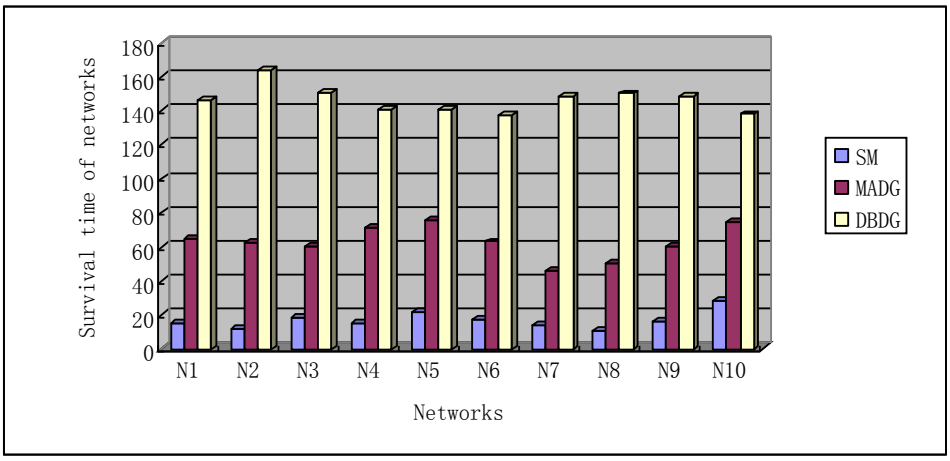

Figure 5. Comparison of network lifetime achieved by the three algorithms

Total energy consumption is the energy consumed by the whole network during its operation. This metric can reflect the degree how sensors' energy is utilized. The result is shown in figure 6. The result indicates that DBDG notably improves this metric. Using DBDG, total energy consumption is improved by 462.78% on average compared to SS, and 197.41% on average compared to MADG.

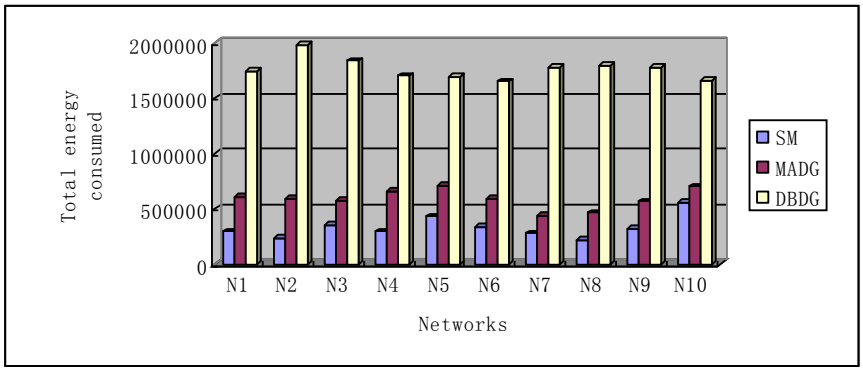

Figure 6. Comparison of network lifetime achieved by the three algorithms

6. CONCLUSIONS

In this paper, we have proposed a novel algorithm for sing mobile sink based on dynamic buffer zone. The algorithm partitions the whole network area into some areas, and let the areas act as the buffer zone in turn. To obtain the reasonable time each area acts as the buffer zone, linear programming is adopted. As a result, the sensors' energy is fully utilized and network lifetime is notably prompted.

ACKNOWLEDGEMENTS

Project supported by the National Natural Science Foundation of China (Grants No. 60703118, 60974082), the Fundamental Research Funds for the Central Universities (Grants No. JY10000970013).

REFERENCES

[1] M. Kohvakka, J Suhonen, M Kuorilehto, V Kaseva, M Hannikainen, T. D. Hamalainen (2009). Ad Hoc Networks, Vol. 7, No. 1, pp24-41.

[2] C.M. Liu, C.H. Lee, L. C. Wang (2007). Distributed clustering algorithms for data-gathering in wireless mobile sensor networks, Journal of Parallel and Distributed Computing, Vol. 67, No. 11, pp1187-1200.

[3] Akyildiz IF, Su W, Sankarasubramaniam Y, etc (2002). Wireless sensor networks : A survey, Computer Networks, Vol. 38, No. 4, pp393-422.

[4] D. Estrin et al. (2009). Next Century Challenges: Scalable Coordination in Sensor Networks, in: The Proceedings of the Fifth Annual International Conference on Mobile Computing and Networks, Seattle, Washington, August 1999.

[5] Ioannis C, Athanasios K, Sotiris N (2008). Efficient data propagation strategies in wireless sensor networks using a single mobile sink, Computer Communications, Vol. 31, No. 5, pp896-914.

[6] Shi Gaotao, Liao Minghong (2007). Movement-Assisted Data Gathering Scheme with Load-Balancing for Sensor Networks, Journal of Software, Vol. 18, No. 9, pp2235-2244.

[7] Roy. Sumit, Jain. Sushant, Brunette. Waylon (2003). Data MULEs: Modeling and analysis of a three-tier architecture for sparse sensor networks, Ad Hoc Networks, Vol. 1, No. 2-3, pp215-233.

[8] Luo J, Hubaux J P (2005). Joint mobility and routing for lifetime elongation in wireless sensor networks, proceeding of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies.

AUTHORS

Zhi Huang received M.S. degree in Information Management from Sichuan University, China in 2006. He is currently a Ph.D. candidate of Computer Science at Xidian University. His research interest is wireless sensor networks.

Sanyang Liu received his Ph.D. in Mathematics from Xi'an Jiaotong University, China in 1989. He is currently a professor in the Department of Applied Mathematics at Xidian University. His research interests include optimization, network algorithms, network reliability and wireless sensor networks.

Xiaogang Qi received his Ph.D. in Mathematics from Xidian University, China in 2005. He is currently an associate professor in the Department of Applied Mathematics at Xidian University. His research interests include high-performance networks and wireless sensor networks.