

A Reduce Identical Event Transmission Algorithm for Wireless Sensor Networks

Hong-Chi Shih¹, Shu-Chuan Chu², John Roddick², Jiun-Huie Ho³,
Bin-Yih Liao¹, and Jeng-Shyang Pan¹

¹National Kaohsiung University of Applied Sciences, Taiwan

²Flinders University of South Australia Adelaide, Australia

³Cheng Shiu University, Taiwan

hqshi@bit.kuas.edu.tw

Abstract. This paper proposed a Reduce Identical Event Transmission Algorithm (RIET). The algorithm can decide that which sensor nodes could send the event to sink node when sensor nodes sense a same even. Moreover, other nodes can save power because they didnt send the same event. In our simulation, the RIET algorithm can enhance sensor nodes life time about 12.9 times and saving power consumption about 52.43% than tradition algorithms.

Keywords: Wireless Sensor Network, Sensor Nodes, Reduce Identical Event Transmission Algorithm, Grade Diffusion Algorithm

1 Introduction

Recent advances in micro processing, wireless and battery technology, and new smart sensors have enhanced data processing, wireless communication, and detecting capability. In general, sensor node is a small and cheap device. Hence, the wireless sensor network is usually constituted by hundred or thousand nodes. In the wireless sensor network, sensor nodes might detect a same event because they were deployed in a large-scale randomly. The sensor nodes will consume lots of power if they send the same event to sink node independently in the tradition algorithm.

This paper proposed a Reduce Identical Event Transmission Algorithm (RIET). The algorithm can decide that which sensor nodes could send the event to sink node when sensor nodes sense a same even. Hence, the RIET algorithm can solve the problem of sending a same event repetitively and reduce sensor nodes power consumption when the sensor nodes detect area are overlapping and sense a same event. Moreover, the algorithm will avoid the transmission delay when sensor nodes commute each other.

2 Related work

This paper proposed the RIET algorithm which is based on Grade Diffusion algorithm (GD) [1] to reduce the power consumption when sensor nodes sense

a same event. We introduce some tradition routing algorithms and the Grade Diffusion algorithm as below.

2.1 Ad-hoc On-demand Distance Vector(AODV)

The AODV was proposed by C. E. Perkins and E. Royer in 1999 [5], and it applies for sensor nodes are fixed or can be moved. The AODV algorithm employs three processes: route request, route replying and route maintenance, to build and maintain the routes required by active nodes for data transmission. The AODV algorithm creates the routing tables for sensor nodes which were on the transmission path when the even appeared, but the routing table just a part of the wireless sensor network. After, the routing table is rebuilt again and consumes lots of power when other nodes have to transfer event.

Please always cancel any superfluous definitions that are not actually used in your text. If you do not, these may conflict with the definitions of the macro package, causing changes in the structure of the text and leading to numerous mistakes in the proofs.

2.2 Direct Diffusion Algorithm(DD)

C. Intanagonwiwat et al. [4] introduced the directed diffusion (DD) protocol in 2003. DD aims to reduce the data relay in order to better manage power consumption. Basically, DD is a query-driven transmission protocol. The collected data are transmitted only if they fit the query from the sink node, thereby reducing the power consumption due to data transmission. In DD, all sensor nodes are bound into a route when broadcasting the interested queries, even if the route is such that it will never be used. In addition, several circle routes, which are built simultaneously when broadcasting the queries, result in wasted power consumption and storage.

2.3 Grade Diffusion(GD)

In 2001, [1] proposed a Grade Diffusion algorithm which improved the LD-ACO algorithm [3] to enhance nodes life time, raised transmissions efficiency and solve the problem of nodes routing and power consumption. The GD algorithm broadcast grade creating packages from sink node to every node completely and quickly in the wireless sensor network by LD-ACO algorithm. Then, the GD algorithm proposed a routing algorithm to reduce nodes average load, enhance nodes life time and, reduce nodes power consumption.

But, the GD algorithm and tradition algorithms just think about that there is just a sensor node sent an event to sink when there is an event appeared. Hence, sensor nodes will consume lots of power to send a same event if they sense a same one. So, this paper proposed a Reduce Identical Event Transmission Algorithm (RIET) which can reduce the probability of sending a same event and saving sensor nodes power. Moreover, the RIET algorithm is based on GD algorithm.

3 Reduce Identical Event Transmission Algorithm(RIET)

The RIET algorithm which is used in the wireless sensor network can reduce the probability of sending a same event, saving nodes power, and enhance sensor nodes life time by sensor nodes communication. The sensor node is not only has the ability of sensing and transferring event but also can commutation with neighbor nodes when it sensors an event. The RIET algorithm uses the finite state machine (FSM) which has Sensing State, Delay State, Query State, and Receive Query State to avoid sensor nodes commutation with neighbor nodes at the same time in the algorithm. Hence, sensor node can query or response neighbor nodes by our algorithm and FSM, and the FSMs transition is showed on Fig. 1.

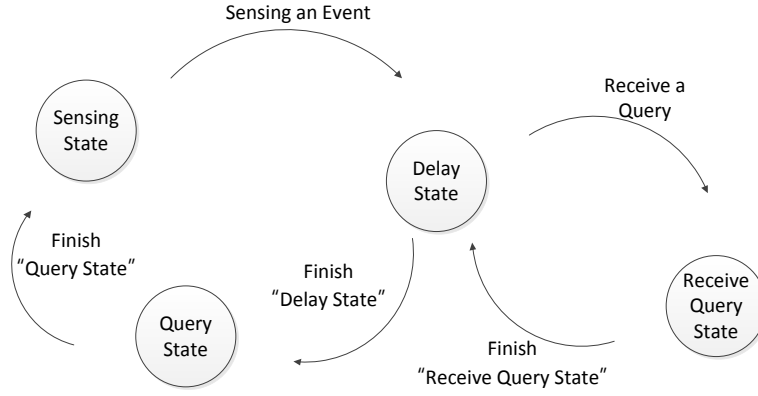


Fig. 1. RIET algorithms final state machine

In Fig. 1, the sensor node has 4 states which are Sensing State, Delay State, Query State, and Receive Query State. In general, sensor node is under the Sensing State process to wait event appeared. When sensor nodes sense an event, sensor nodes state will be changed to Delay State and each node which senses the event will get a different delay time W_d . The W_d illustration is in section 3.1. Then, the sensor nodes will be change to the Query State after the delay time. If sensor nodes state is Query State, they will commutation with neighbor nodes. Hence, the delay time can avoid the sensor nodes change their state to Query State at the same time, and then send event to sink node independently.

After Delay State, sensor nodes state will be change to Query State and send a query message to neighbor nodes. The message is asking neighbor nodes whether or not you sense a same event. The sensor node will be changed to Receive Query State when node is under the Delay State processing and receives a query message. If the sensor nodes delay time is short than other nodes, the sensor nodes state will be changed to Query State faster than others. Then other neighbor nodes will receive a query message and change their state to Receive

Query State from Delay State. The illustration of Query State and Receive Query State is in the session 3.2 and 3.3.

3.1 Delay State

After sensor node senses an event, the node will record an event processing state E_s and the E_s has three states: Un-processing (-1), Processed (+1), and In-processing (0). The Query State and Receive Query State will change the E_s , and their illustrations are in the session 3.2 and 3.3.

The W_d is calculated by equation (1), and it is a delay time mean when the sensor node can change its state from Delay State to Query State.

$$W_d = \left(\alpha + N_l + \left(1 - \frac{E_{now}}{E_{original}} \right) + (1 - \alpha) * P_r \right) * T_d$$

$$N_l = \begin{cases} \left(\frac{l_v - 2}{l_v} \right), & l_v > 2 \\ 0, & otherwise. \end{cases} \quad (1)$$

In equation 1, α is a user set value between 0 to 1, T_d means the shortest delay time, P_r is a random value between 0 to 1, l_v means sensor nodes grade value, E_{now} means sensor nodes remaining power, and $E_{original}$ means sensor nodes initial power.

In the RIET algorithm, sensor node needs to wait an $\alpha * T_d$ delay time least. And, if sensor node's E_{now} is more than others, the sensor node's state can change to Query State quickly. Moreover, the sensor node whose grade value is smaller than others will have shorter delay time W_d than others because the event might be sensed by different grade sensor nodes.

3.2 Query State

In Fig. 1, the sensor node's state will be changed to Query State from Delay State after delay time. When a sensor node is under Query State processing, the sensor node will check its E_s firstly. If the E_s is Processed (+1), the sensor node will do nothing and change its state to Sensing State because the event has been processed by other nodes.

On the other hand, the sensor node will query its neighbor nodes because the event hasn't been processed if sensor node's E_s is Un-processing (-1) or In-processing (0). The sensor node which sent query messages sets a minimum waiting time T_w to wait neighbor nodes' response, and the sensor node will check received information which contains E_s and W_d from neighbor nodes.

The query node check its and response node's E_s . If the response node's E_s is Processed (+1), the query node will do nothing and change its state to Sensing State because neighbor nodes had sent event to sink node.

But, if response node's E_s is Un-processing (-1) or In-processing (0), the query node will check its and response node's W_d . The query node sends the event to sink node, modify its E_s to "Processed (+1)", and change its state to "Sensing State when its W_d is smaller than response node's. On the other hand,

the query node just changes its state to “Sensing State when its W_d is bigger than response node’s or equal because neighbor nodes will send event to sink node after.

3.3 Receive Query State

The sensor node’s state will be changed to “Receive Query State from “Delay State when it received a query message.

When the sensor node is under the “Receive Query State processing, it will check whether or not it had sensed a same event as the query node. The sensor node will do nothing and change its state to “Delay State if it didn’t sense the same event. On the other hand, the sensor node responses its E_s and W_d to the query node, and receives the query node’s E_s and W_d if the sensor node has the same event.

When the sensor node got the query node’s W_d and E_s , it will check them. If the sensor node’s E_s is “Processed (+1), it will do nothing and change its state to “Delay State. Otherwise, the sensor node will check its W_d when its E_s is “Un-processing (-1) or “In-processing (0) because the event hasn’t been sent. If the sensor node’s W_d is smaller than the query node or equal, the sensor node changes its E_s to “Processing (0) and its state to “Delay State because the sensor node might send the event to sink node after. The query node will change its E_s to “Processed (+1) and finish its “Query State if its W_d is bigger than the receive node’s.

The sensor node is still under the “Delay State till the delay time W_d end when it returns from “Receive Query State” processing. If the sensor node which is under the “Delay State” processing and it receives a query message again, the sensor node’s state will be changed to “Receive Query State” again.

Additionally, the sensor node won’t into the “Query State” and “Receive Query State” processing at the same time because we need to avoid they change E_s at the same time.

4 Simulation

We simulate and analyze the RIET algorithm in this session. In the simulation, we build a 3-D space of $100*100*100$, and the scale of the coordinate axis for each dimension is limited from 0 to 100. The radio limit of the nodes was set to 15 units. There are three non-overlay nodes in each $10*10*10$ space and their Euclidean distance each other is at least 2. Thus, the space includes 3000 nodes, and the center node in (50,50,50) is assigned to the sink node.

There are 300 cycles and each cycle has 300 event been appeared. Thus, there are 90000 events in our simulation. Every sensor node’s initial power is 3600mw and it will consume 1.6mw to send an event. Moreover, we set the parameter α is 0.8 in equation 1 and the E_{th} is 10%. In the simulation, the sensor nodes have a same event if the event’s values are different less than E_{th} .

Firstly, we compare the active nodes of the DD, GD, and RIET algorithm in each cycle, and the result was showed on Fig. 2. And, the active node means that it has enough power to send event and has relay nodes in its routing table.

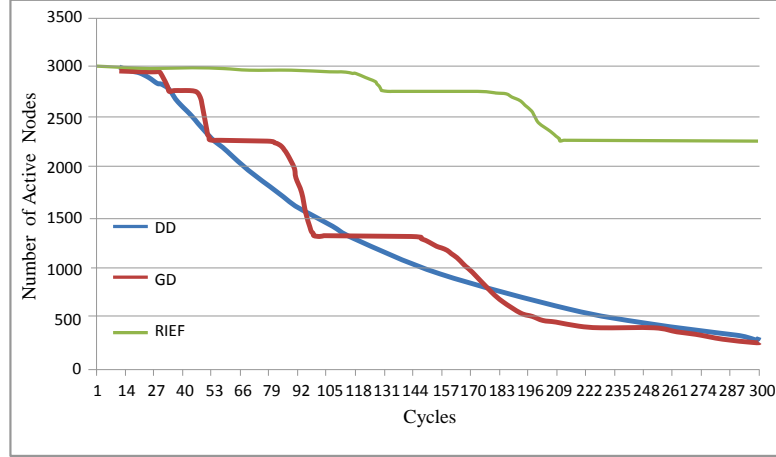


Fig. 2. Number of active nodes

In Fig. 2, the DD and GD algorithm only have 303 and 176 active nodes after 300 cycles, but the RIET still has 2240 active sensor nodes which has power can sense and send event. The RIET algorithm rises sensor node's lift time about 7.5 times and 12.9 times than the DD and GD algorithms because the RIET algorithm can reduce the sent numbers of sensor nodes which sense a same event. But the sensor nodes will send an event to sink nodes independently if they sense the same event in the DD and GD algorithms, and they consumed lots of power. Hence, the RIET algorithm can save lots of power and rises sensor node's lift time. So, the RIET algorithm's active nodes are more than the DD and GD algorithms in Fig. 2.

Then, we compare the average power consumption of the DD, GD, and RIET algorithms, and the result was showed on Fig. 3. In Fig. 3, we can find that the RIET algorithm has least average power consumption because it can reduce the sent numbers of sensor nodes which sense a same event. Hence, the average power consumptions of the DD and GD algorithms are 3497.83mw and 3342.16mw after 90000 events appeared, but the RIET algorithm only consume 1663.94mw. So, the RIET algorithm can reduce average power consumption about 52.43% and 50.21%.

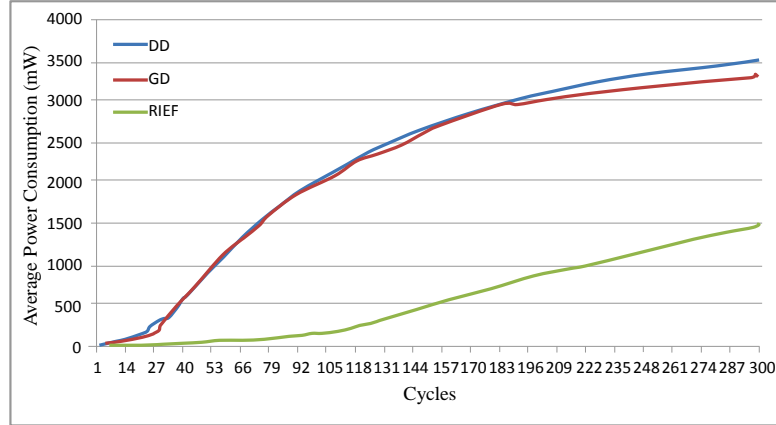


Fig. 3. Sensor nodes' average power consumption

5 Conclusion

This paper proposed a Reduce Identical Event Transmission Algorithm (RIET). The algorithm can reduce to send a same event repeatedly, raise sensor node's lift time, and reduce sensor node's power consumption.

In our simulation, we can find that the RIET algorithm can save sensor node's power about 52.43% and 50.21% than tradition algorithms. Moreover, the RIET algorithm can enhance sensor node's lift time about 12.9 times.

References

1. Shih, H.-C., Ho, J.-H., Liao, B.-Y., Pan, J.-S.: Grade Diffusion Algorithm. In: Conference on Information Technology and Application in Outlying Islands 2011
2. Corson, S., Macker, J.: Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. RFC 2501, 1999
3. Ho, J.-H., Shih, H.-C., Liao, B.-Y., Chu, S.-C.: A Ladder Diffusion Algorithm Using Ant Colony Optimization for Wireless Sensor Networks. In: Information Sciences, 2011
4. Intanagonwivat, C., Govindan, R., Estrin, D., Heidemann, J., Silva, F.: Directed Diffusion for Wireless Sensor Networking. In: IEEE/ACM Transactions on Networking, vol. 11, pp. 2–16, February 2003
5. Perkins, C. E., Royer, E.: Ad Hoc On-Demand Distance Vector Routing. In: Proceedings of IEEE WMCSA, pp. 90–100, 1999
6. Royer, E. M., Toh, C.-K.: A Review of Current Routing Protocols for Ad-Hoc Mobile Networks. In: IEEE Personal Communications, vol. 6, pp. 46–55, 1999