# Managing Collaborative Protocol for Wireless Sensor Networks

Thasayu Soisoonthorn and Sirichai Rujipattanapong
Department of Electrical Engineering, Faculty of Engineering
King Mongkut University of Technology North Bangkok, Bangkok 10800 Thailand
Tel: +66-2-9132500 ext 8519, Fax: +66-2-5857350
E-mail: srujipat@kmutnb.ac.th

***Abstract*— Wireless sensor network protocols are mostly designed with each node independently processing and transmitting small amount of data to base station. Thus, they may be unsuitable for some applications, such as target tracking. In this kind of application, it requires data from many nodes to track target accurately. Hence, transmitting such large data directly to base station is not efficient. It can cause contention, collision, and reduced performance of the whole network. Thus local collaboration among nearby nodes is necessary.**

**In this paper, the protocol that helps manage the collaboration among neighbor nodes is proposed. This protocol is named managing collaborative protocol (MCP). Based on the modification of TTDD, its design goal is to reduce the overhead of cluster formation. Compared to both DCTC and BACKOFF, using ns2, MCP reduces the number of transmissions; delivery success rate is higher, and area coverage of the event is better.**

## I. Introduction

Manufacturing technology of electronic devices improves at a tremendous rate. As a result, sensor nodes are more capable and consume less energy. However, sensor node still cannot harvest energy directly from environment. Therefore, energy consumption must be taken in to account when designing protocol for sensor nodes. Thus, it is preferable to design protocol that can manage collaboration and balance energy utilization among sensor nodes to achieve the objective of the application.

Related works are summarized here. In UW-API [1] and Dynamic Clustering [2] both use collaboration. However, they both have shortcomings. That is, each node has specific role: cluster head and ordinary node. Although it is easy to form cluster, once deployed, node cannot change its role. Thus if anything happen to cluster head, that particular cluster lose communication with the rest. In BACKOFF [4], using broadcast, the formation of cluster is achieved by contention among neighboring nodes. However, there may be too many cluster heads created; Data sent to base station are then in a great amount. And it can expend too much energy if the base station is far away from the detecting area. Moreover if the communication range is low or few sensor nodes are in the area, then its effectiveness is affected.

Assuming all nodes are ordinary, DCTC effectively create cluster by using GAF [6] to limit the number of nodes used in forming cluster. However, due to its use of flooding, the number of collisions increases. Even though it uses GAF to reduce flooding, since the grid size of GAF depends on the sending range, thus if the sending range is short or the detection area is large, the number of packets used in forming cluster is high, causing increased collision and expending more energy.

From the above mentioned problems, this paper proposes to improve the cluster forming for collaborative processing. Based on the modification of TTDD, nodes can calculate the position of grid head without needing any additional packet except hello packet. Sensor node that is closest to this position is grid head. There may be more than one grid head. When sensor node detects event, it sends information to the closet grid head. After that, all grid heads start the process of forming cluster by electing cluster head. Cluster head then has the responsibility to coordinate collaboration among grid heads and summarize the report of target location to base station.

From the simulation result using ns2, it is found that MCP consumes less energy than DCTC about 19.76% in the large detecting area. When detecting area is small, MCP expends energy about the same as DCTC. In addition, its coverage area is higher than DCTC. BACKOFF [4] has lower control packet transmission. The number of packet drops is also lower.

In this paper, the modification of TTDD for cluster creation is described in Section 2. The detail operation of MCP is explained in Section 3. Section 4 shows the result and compare with other protocols. Finally, the conclusion is given in Section 5.

## II. The Modification of TTDD

The details of TTDD operation can be found in [10]. The main different features are summarized as shown in Table 1.

Table 1. The main different features of MCP and TTDD

| Features | MCP | TTDD |
|---|---|---|
| Grid Intersection | Globally created | Created for each source |
| Balance Energy | Rotating by time | Based on position of sensor |
| Cluster Head | Contention Mechanism | No Cluster Head |
| Path Building | Only in cluster | Between sink and source |
| Collaboration | Exist in one cluster | No |
| Tracking | Yes | No |
| Aggregation | Yes | No |

## III. THE OPERATION OF MCP PROTOCOL

Sensor nodes that detect the event are source nodes. Node that collects data from sensor nodes within the grid is called "Grid Head." And sensor node that collects all data from grid heads is called "Cluster Head." Each sensor node can be source, sink, grid head, or cluster head. Thus, when user wants to know the data of interest event, the sink node is set to broadcast event interest to sensor nodes. In this paper, base station is assumed to have unlimited power supply, and can communicate with every sensor node in the whole network [8]. By this assumption, each node knows the destination location, and event to detect. When sensor nodes detect the event, all data are then sent to grid head. After that, grid head collects and summarizes all raw data and transmit to cluster head. Cluster head then summarizes all data received from every grid head and then send to base station. Algorithm in forming grid head, cluster head, and path to cluster head is the main work and contribution of this paper.

### *A. Data Component of the sensor nodes.*

In each sensor node, it deals with two types of data: event detection data, and route selection data. Data stored at each sensor node are node ID, position ($x,y$), grid ID, neighbor list information, e.g., neighbor node position, node ID, information about destination node, e.g., destination node position, event to detect, and information of cluster head, such as grid ID it resides, energy level sensed by cluster head, position, and the priority rank of being cluster head.

### *B. Type and Component of Packets.*

Packets used in selecting path are three types as below.

1. Hello Packet. This packet is 18 bytes in size. It consists of packet type, sensor node number, position. By sending hello packet, node can know about its neighbor nodes within communication range. When node receives the hello packet, it updates its own table.
2. Control Packet. There are two types. First type is used to announce current cluster head. This packet contains type of packet, grid number it belongs to, energy detected, cluster head rank. Second type is packet sent by cluster head to grid head node. This is to ask that grid head to be new cluster head. It consists of type of packet, grid number intended, and the rank of cluster head.
3. Data Packet. There are two types of data packet. First, it is raw data. This data is detected by sensor node and sent to grid head. Its component consists of type of packet, position of node able to detect event, energy level detected, type of signal, and detection time. Second is the processed data. Then grid head send this processed data to cluster head. And cluster head summarizes and finally send to base station. The processed data contains type of packet, node ID, sequence number, hop counts, type of detected event, position and time of detection.

### *C. Algorithm of Cluster Formation.*

#### *C.1 Cluster Forming Process*

To form the cluster, each sensor node in the detecting area needs to have the ID and position of neighboring nodes. Thus every sensor node identifies itself by sending hello packet and then extracts all information of its neighbors from hello packets received. In this paper, the grid is created throughout the whole network. Grid intersection is assigned to be the point where Grid Head resides. However, due to the irregularity of the network, node that is closest to the grid intersection is the Grid Head. Grid intersection can be calculated using Equation 1 and 2. This grid intersection is global, since every sensor node also can calculate this grid intersection, if it has the same referred position and time. Therefore, when target enters the network area and sensor node can detect it, sensor node then locate itself closet to which grid head. After that, it sends data to that grid head using Greedy Forwarding [7]. When there are no other nodes closer to grid intersection than itself, then that node is grid head.

$$x_i = i \times \alpha + \beta_x \qquad (1)$$

$$y_j = j \times \alpha + \beta_y \qquad (2)$$

where $x$ and $y$ are the coordinate of grid head, $\alpha$ is the grid size, and $\beta_x$ and $\beta_y$ are the timers used to rotate grid head to avoid overload. Both can be calculated using Equation (3) and (4), respectively.

$$\beta_x = \Delta d \times \mathrm{mod}\left( \left\lfloor \frac{NOW}{round} \right\rfloor, \left\lfloor \frac{\alpha}{\Delta d} \right\rfloor \right) \qquad (3)$$

$$\beta_y = \Delta d \times \mathrm{mod}\left( \mathrm{mod}\left( \left\lfloor \frac{NOW}{round} \right\rfloor, \left\lfloor \frac{\alpha}{\Delta d} \right\rfloor \right), \left\lfloor \frac{\alpha}{\Delta d} \right\rfloor \right) \qquad (4)$$

where $NOW$ is the current time, $round$ is the waiting time before changing position in each round. In sending to each grid head, grid head number is not used since it can be changed. Instead, grid head is referred to as grid number as shown in Equation (5).

$$GID = i \times \left\lceil \frac{MAXX}{\alpha} \right\rceil + j \qquad (5)$$

where $GID$ is grid number, $MAXX$ is the network size in x-axis and $\alpha$ is the grid size.

As can be seen, the grid head is easily established without sending any control packets. However, target may move throughout several grid areas, this cause many grid heads being created. These grid heads then each send data to base station several times, wasting precious energy. Further, data received from sensor nodes within grid may be not sufficient to process. Thus grid heads have to communicate with each other to elect the cluster head responsible to collaborative processing of the event.

Therefore, when grid head received data from sensor nodes in its own grid area, if it does not have information of who is the cluster head, it establishes itself as cluster head and then informs its neighbor nodes. When there are several grid heads contend for cluster head, only grid head nearest to the event, judged on the detected energy, can be the cluster head. Cluster head contention is only carried out by grid heads able to detect the event; others ignore this contention.

### *C.2 Cluster Maintenance*

When the cluster head election process is completed, it may be possible that, if there is other detections, other grid heads outside the cluster detect the event and proclaim themselves as cluster head. To prevent this circumstance, grid heads in the current group store the Time Stamp of cluster head election, to show the duration the group is created. Thus, when new grid head announces itself as cluster head, and when the grid head that are in the current cluster receive this packet, it can find out that the new cluster head has newer time stamp, then it will send out to this new proclaimed cluster head to inform of the current cluster head.

After the cluster head has reported data to base station, target may move out of its current position and grid head that resides out of this cluster can detect the target and establish itself as cluster head again. To prevent this kind of problem, current cluster head will predict the location of target using [11] and identify the grid head that likely detects this target. Then it sends packet to inform the identified grid head of the current cluster head before the target entering that grid area.

### *C.3 Target Tracking and Cluster Head Rotating*

When cluster head evaluates that the position of the target is farther than the specified distance, causing that data be sent over long distance before reaching cluster head, it sends out packet to inform the grid head closet to the target and ask it to be the new cluster head. When that grid head accepts, it announces this to all grid heads.

## IV. SIMULATION RESULTS

In this section, the effectiveness in forming cluster in wireless sensor network is presented. Comparison is made with BACKOFF[4] and DCTC[5] with prediction based and localized reconfiguration. Using ns-2, the interested metrics are:

1. Average energy used by sensor nodes
2. The number of Cluster Head created. If this number is high, it wastes too much energy sending data to base station. And if the base station is distant, it aggravates the energy usage. Furthermore, its coverage percent is low.
3. Percent coverage. The ratio of the number of nodes belonging to cluster to the total number of nodes that can detect event. If the number of members is high, then the data needed for processing are large, resulting in better quality data and higher accurate.
4. The number of control packets. This is the number of packets used in forming the cluster.
5. The total number of packets sent. This includes data packet, raw data, control and others. If this number is high, the energy is also highly expended. Also the number of contentions is high, causing high collision.
6. The number of drop packets. This is the total drop packets in the whole network.

Simulation environment: The area of detection is set to be 1000×1000 $m^2$. The number of nodes is 400. Node distribution is random. Energy model in [12] is used in the packet transmission. Data transmission model uses Event-Radius Model (ER) [9]. Target is randomly originated in the whole network area. Target can move randomly with the maximum speed of 5 m/s.

The experiment is set to test the algorithm by setting the communication range of node to be 100, 150, 250, and 300 meter. There is only one target present at any time. The event size is 500 m. The grid size of MCP is 200 and 500 m.

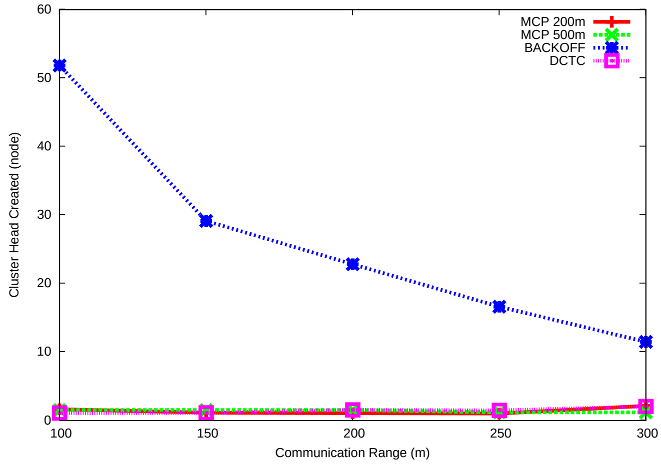


Figure 1. The number of Cluster Head Created for event size 500 m.

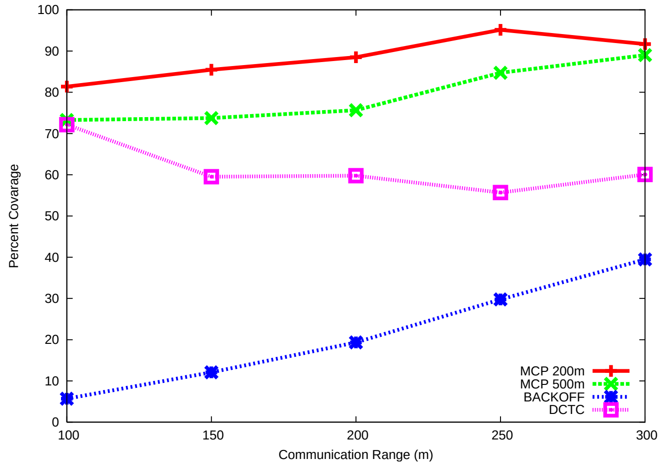


Figure 2. Percent Coverage for event size 500 m.

From Figure 1, which shows the number of cluster heads created, it can be seen that BACKOFF has the highest number of cluster heads. This is due to the fact that BACKOFF uses only transmission range in forming cluster. If this range is short, the number of cluster heads created is then increased. This fact also affects the percent coverage in Figure 2, which can be seen that BACKOFF has very low percent coverage. For DCTC, the percent coverage is 65 %, while MCP has the highest coverage.

In Figure 3, the energy usage of DCTC is highest, owing to its uses of flooding. Also, since the communication range is shorter than the detection area, thus the number of grid heads is high. In the case of MCP with 500m grid size, the energy usage is high in the short transmission range. The reason for this is that when its sending range is short, it needs multiple hops before it reaches grid head. When the range is larger, MCP 500m grid size consumes less energy than DCTC. At the range size of 300m, MCP expends energy 19.76% less than DCTC. BACKOFF, in this case, consumes about the same amount of energy as MCP. But from Figure 1, since the number of cluster heads is highest, it should consume high energy, too. This is because in this experiment, the size of network is not large. The size cannot be set large because the memory limitation of ns2. If the distance between cluster head and base stations is close, energy usage then is low. Since it sends raw data directly to base station, when the base station is near, it takes less energy. On the contrary, since other protocols need to send data to cluster head, it consumes more energy. Moreover, BACKOFF does not have target tracking, thus it can avoid sending large number of control packets.

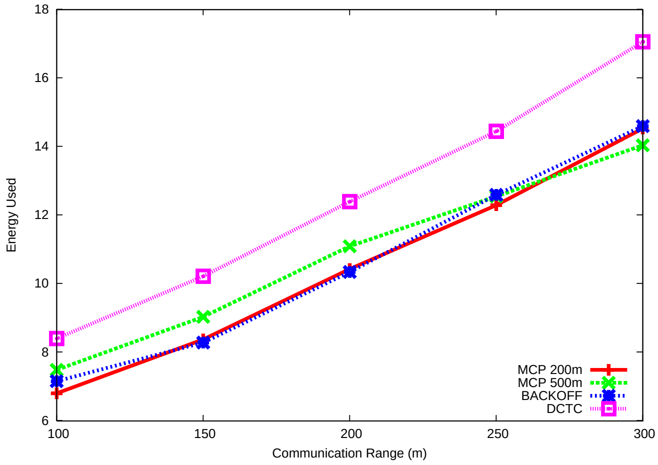


Figure 3. The energy used per node for event size 500m.

Figure 4 shows the number of control packets. MCP with larger grid size requires less control packets but it may increase the distance of data transmission and the detection range. BACKOFF has the number of control packets higher than MCP, with 500m grid size, about 13.7 times.

Figure 5 shows the total number of packets sent when the target range is 500 m.. It can be seen that DCTC has the high number of packets sent. This is because nodes need to send packets to cluster head. While MCP collect data first and summarize before sending to grid head. Thus the number of packets sent is lower. And if the grid size is set too large, it increases the distance the data need to travel before reaching cluster head, thus wasting high energy.

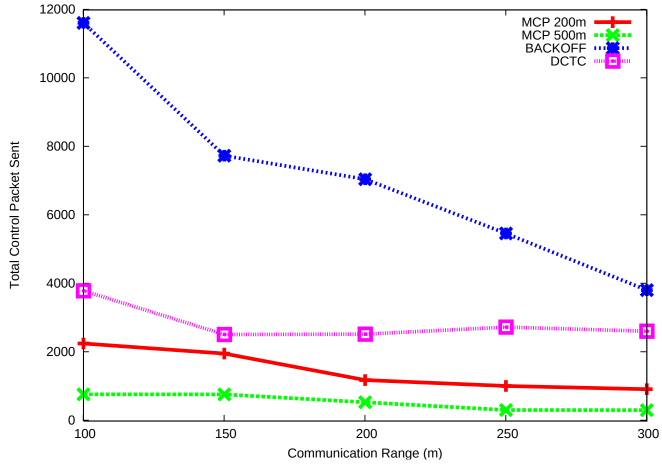


Figure 4. The total number of control packets sent for event size 500 m. and 400 nodes.

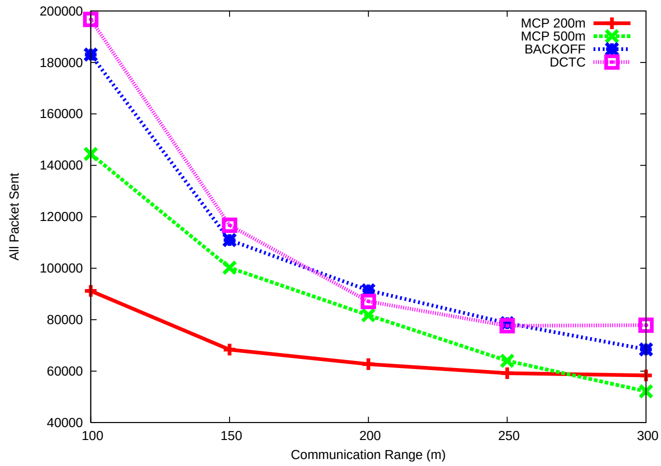


Figure 5. The total number of all packets sent for event size 500 m. and 400 nodes.

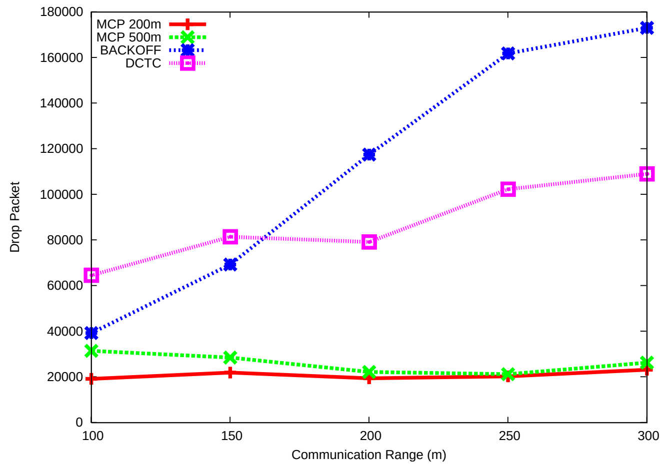


Figure 6. The total number of drop packets for event size 500 m. and 400 nodes.

Figure 6 shows the total number of drop packets in network for the same target range. It can be seen that both DCTC and BACKOFF has high number of packets dropped. This is because the use of broadcast, causing high collision. And this collisions highly depends on the transmission distance. If the distance is high, the probability of collision is also high. For MCP, since it uses unicast and the total number of packets is low, thus the number of collisions is low, independent of communication range of sensor node. However, when MCP grid size is large, the drops will increase slightly. This is because of the distance it send is far and the transmission range is low, causing multiple hops needed. Thus the drops are higher.

## V. CONCLUSIONS

Energy usage is the main concern for wireless sensor network. Thus if sensor nodes send data directly to base station, it wastes too much energy. Using collaboration can help reduce energy consumed. However, the overhead of creating cluster may be high. In this paper, the overhead reduction is main focus. From the experiment, it shows that the energy usage in MCP is 19.76% less than DCTC. And the number of control packets required to build cluster is very low, resulting in low collision. In addition, the experiment result shows that the transmission range has less impact on MCP than DCTC and BACKOFF.

## ACKNOWLEDGMENT

Authors would like to thank the King Mongkut University of Technology, North Bangkok (KMUTNB), for the overseas presentation funding.

## REFERENCES

[1] P. Ramanathan, K. Saluja, K.-C. Wang, T. Clouqueur, "UW-API: A Network Routing Application Programmer's Interface." *Technical Documentation for DARPA SensIT Program*, October 2001.

[2] Chen, W.P., Hou, J., Sha, L., "Dynamic clustering for acoustic target tracking in wireless sensor networks", *11th IEEE International Conference on Network Protocols.* (2003) 284 - 294

[3] G. Wang, W. Zhang, G. Cao, and T. La Porta., "On Supporting Distributed Collaboration in Sensor networks", *IEEE Military Communications Conference (MILCOM)*, October 2003.

[4] Jeongkeun Lee, Kideok Cho, Seungjae Lee, Taekyoung Kwon, Yanghee Choi., "Distributed and energy-efficient target localization and tracking in wireless sensor networks", *Computer Communications* 29(13-14): 2494-2505 (2006)

[5] W. Zhang and G. Cao., "DCTC: Dynamic Convoy Tree-Based Collaboration for Target Tracking in Sensor Networks", *IEEE Transactions on Wireless Communication*, September 2004.

[6] Ya Xu, John Heidemann, and Deborah Estrin., "Geography-informed energy conservation for ad hoc routing", *Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking*, Rome, Italy, July 2001. ACM.

[7] Karp, B., and Kung, H. T. "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks." *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking* (MobiCom) Boston, MA, Aug. 2000.

[8] L. Zou, M. Lu, and Z. Xiong., "PAGER-M: A novel location-based routing protocol for mobile sensor networks", *Proc. BroadWISE'04*, San Jose, CA, October 2004.

[9] B. Krishnamachari, D. Estrin, and S. Wicker. , "Modelling Data-Centric Routing in Wireless Sensor Networks", *USC Computer Engineering Technical Report* CENG 02-14, 2002.

[10] F. Ye, H. Luo, J. Cheng, S. Lu and L. Zhang.,"A Twotier Data Dissemination Model for Large-scale Wireless Sensor Networks", *ACM/IEEE Mobicom* 2002

[11] A. Aljadhai and T. Znati., "Predictive mobility support for QoS provisioning in mobile wireless environments", *IEEE J. Select. Areas Commun*., vol. 19, pp. 1915–1930, Oct. 2001.

[12] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. *Proceedings of the Hawaii International Conference on System Sciences*, Maui, Hawaii, January 2000.

[13] " The network simulator - ns-2." [online] [cited 2003 Nov 23]., Available from : URL : http://www.isi.edu/nsnam/ns/.