# A Data Gathering Mechanism based on Clustering and In-Network Processing Routing Algorithm: CIPRA

Eunhyoung Chu[*] Tsunenori Mine[**] Makoto Amamiya[**]

Department of Intelligent Systems,
{[*]Graduate School , [**]Faculty} of Information Science and Electrical Engineering, Kyushu University
744 Motooka, Nishi-ku, Fukuoka 819-0395 JAPAN
E-mail: [*]chueh@al.is.kyushu-u.ac.jp [**]{mine, amamiya}@is.kyushu-u.ac.jp

## ABSTRACT

Sensor networks require energy-aware, efficient data-collecting methods to extend their network lifetime. In this paper, we propose an energy-efficient data gathering mechanism which clusters sensor nodes and forms a distributed data-routing tree based on in-network data fusion. In our mechanism, the cluster formation and the data-routing tree construction are simultaneously carried out so that they reduce their energy required to organize a multi-hop routing tree of sensed data. The mechanism also performs data aggregation at each member node to reduce the amount of transmission data. Moreover our work distributes energy load to each node to avoid the intensive energy consumption of a cluster-head. Experimental results show that our data gathering mechanism outperforms the direct scheme protocol and the LEACH protocol on the point of view of the network lifetime.

***Keywords***: Sensor networks, data gathering, clustering, data aggregation, data-routing tree

## 1 INTRODUCTION

Recent advances in computing technology have led to the development of a new computing device: the wireless, battery-powered, smart sensors. Sensors which are capable of sensing, computing and communicating may be deployed in ad hoc network environments without infrastructure and centralized control. Self-organizing and self-configuring capability are requisite for such sensor networks. In addition, activities of sensors are severely constrained by limited resources such as battery power, memory and computing capability available, which require sensor networks to be energy-efficient.

Direct transmission that sensors directly transfer data to the base station, may not be adequate to the sensor networks because of their limited power. Accordingly communication should occur via intermediaries in a multi-hop fashion. Moreover because adjacent sensor nodes obtain similar or identical data, using in-network aggregation in a multi-hop communication is useful to reduce the volume of transmission data. A clustering technique which gathers data from several representative sensor nodes by grouping sensors provides scalability for the sensor networks that are composed of hundreds or thousands of nodes. Clustering is essential for applications requiring efficient data aggregation. Another advantage of clustering technique is to reduce energy consumption of the network [2-12].

In this paper, we present an energy-efficient data gathering mechanism which employs a hierarchical clustering algorithm and in-network processing. We call the mechanism CIPRA (data gathering mechanism based on Clustering and In-network Processing Routing Algorithm). CIPRA prolongs network lifetime by distributing energy consumption. CIPRA distributes the energy load of the cluster-head to member sensors so that energy of each sensor equally decreases over the whole network. In CIPRA, after sensors sense data, each node sends the data to its neighbor node instead of its cluster-head. Neighbor nodes aggregate data to reduce the amount of data and transfer the aggregated data to their neighbor nodes, which may be their cluster-head. Moreover CIPRA is able to self-organize a data-routing tree and has self-configuring capability by using local information of each sensor. Sensors should dynamically adjust radio transmission energy to adapt to the change of a network topology caused by disappearance of nodes. Using local communication among neighbor nodes lessens the communication distance. In-network processing at each member node distributes the energy load of cluster-heads to the member nodes.

The rest of the paper is organized as follows. Section 2 discusses related work, and Section 3 describes our sensor networks and a radio model. Section 4 describes the proposed algorithm in detail. And then in Section 5 experimental results show energy-effectiveness of our algorithm. Then we conclude in Section 6 and present the future work in Section 7.

## 2 RELATED WORK

Many protocols have been proposed and designed in order to extend the lifetime of sensor networks with constrained resources. For example, Directed Diffusion is data-centric in its network view and performs all routing decisions through local, neighbor-to-neighbor interactions [8]. It provides a reactive routing technique, discovering routes between information sources and the base station. TAG (Tiny Aggregation) is a generic aggregation service for ad hoc networks of TinyOS motes. TAG aggregates data in the network using piggybacking aggregation queries on the existing ad hoc network protocol [9].

As a representative clustering protocol LEACH protocol was proposed [4-5]. It was a solution using randomized cluster-head selection and data aggregation at cluster heads. In LEACH, a pre-determined percentage of sensor nodes become cluster-heads per round. After clusters are formed, their cluster-heads gather and aggregate sensor data from their members in their vicinity, and transfer the aggregated data to the base station. LEACH employs the cluster-head rotation for balancing the energy load of cluster-heads. However LEACH does not guarantee to make a good cluster head distribution and select the pre-determined optimal number of cluster-heads per round. ACE makes clusters of the senor networks using the node degree as the main parameter [12]. In PEGASIS, each node aggregates data over a chain routing path after forming it with the closest neighbor and only a cluster head transmits the aggregated data to the base station [6]. HEED selects cluster-heads according to nodes' energy and a secondary parameter, such as node proximity to nodes' neighbors or degree of the node [7].

## 3 SENSOR NETWORKS

We assume sensor networks have the following properties

(1) A set of sensor is scattered on a square field (Field*Field).

- Each sensor-node is uniformly deployed over the entire network.

(2) The sensor-nodes are located in a fixed and unknown place.

(3) The sensor-nodes enable to adjust their radio radius (transmission power).

(4) Applications

We consider applications that allow data items of fixed sizes to be fused together and be compressed into a signal. For example, an environment monitoring application that measures average, count, sum, maximum and minimum of environment parameters such as temperature, humidity and brightness is an instance.

(5)Radio model

We assume a node has a few discrete transmission power levels. In this paper we use the same radio energy dissipation model as LEACH. To transmit an l bit message a distance d which means Euclidean distance as physical length, the radio expends

$E_{TX}(l,d)=E_{TX\text{-}elec}+E_{TXamp}(l,d)$

$=l*E_{elec}+l*\varepsilon_{amp}*d^k$ (1)

and to receive this message, the radio expends

$E_{RX}(l)=E_{RX\text{-}elec}=l*E_{elce}$ (2)

In figure 1, there are some communication energy parameters: the electronics energy ($E_{elec}$), the amplifier energy $\varepsilon_{amp}$ and the energy for data aggregation ($E_{DA}$). $\varepsilon_{amp}$ varies according to distance d between a sender and a receiver : $\varepsilon_{amp}=E_{fs}$ assuming a free space model when $d<d_0$ and k=2, while $\varepsilon_{amp}=E_{mp}$ assuming a multipath model when $d\leq d_0$ and k=4, where $d_0$ is a constant distance that depends on the environment. Each parameter is set as follows: $E_{elec}$=50nJ/bit, $E_{fs}$=10nJ/bit, $E_{mp}$=0.0013pJ/bit, $E_{DA}$= 5nJ/bit/signal and $d_0$=75m.

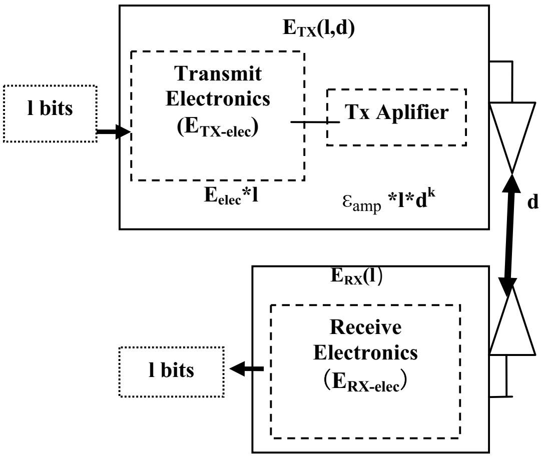


Figure 1: The radio model

(6) Each node can calculate the Euclidean geometric distance to a transmitter by using the received signal strength if transmission power is given.

(7) We ignore the energy wasted during packet collisions as well as start-up transients. Namely, collision does not occur.

## 4 ALGORITHM

### 4.1 Protocol outline

In this section, we describe CIPRA, a novel data gathering algorithm that employs a clustering architecture based on in-network aggregation at each sensor node.

A cycle in which all sensor-nodes transfer obtained data from surroundings to a base station via cluster-heads is called a round. CIPRA periodically rotates the round. Each round begins with ( i ) clustering phase, when a single sensor-node becomes a cluster-head that takes a role in the network as a control node or a representative transfer data to the base station to cluster the network and broadcasts cluster-head announcement to the other nodes, followed by ( ii ) routing setup phase when a multi-hop data-routing tree, rooted a cluster-head, is constructed by broadcasting based on a hybrid medium access control method that consists of CSMA (Carrier-Sensing Multiple Access) and TDMA (Time-Division Multiple Access). And last, data transmission phase when sensor-nodes transfer data to the base station occurs. We call the time spent by clustering phase, clustering time $T_{cluster}$, the time spent for route setup phase, routing setup time $T_{routing}$, and the time spent by data transmission phase, data transmission time $T_{tx\_data}$. To minimize overhead, $T_{tx\_data}$ is much longer than time interval suming $T_{cluster}$ and $T_{routing}$.

In the following, we use the terms a parent node and a child node along a multi-hop data-routing tree. The next hop recipient to which a sender transfers a packet destined for the cluster-head is called its (sender) parent sensor, while to the parent sensor (recipient); this sensor (sender) is its (recipient) child sensor. Each sensor becomes a source and router with the ability of data aggregation over the multi-hop data-routing tree.

Though the sensor-nodes may start the protocol at slightly different times due to network delay or clock discrepancies time synchronization is not required in our data gathering mechanism. Each node initiates its actions at its iteration time with a delicate difference against other sensor-nodes. Since each node corresponds to message exchange, time synchronization does not impact of performance. Subtle time difference between nodes helps to avoid collisions.

We note that time for transferring data is to consider propagation delay and reception and transmission delay. And we assume each phase such as iteration time for clustering and transmission time between tree levels has enough time to perform these tasks[12].

## 4.2 Cluster head selection: A cluster head

Sensors consume considerable energy to send data to a remote base station located in far away distance. To lessen the number of sensors transferring data to the base station may prolong the network lifetime because of reducing energy consumption. It is most energy-efficient to have only a cluster head.

If each node knows a priori the total number of sensors, N, and has its unique identification (ID) which is a number from 0 to N-1, it decides whether or not to elect itself as a cluster-head by computing node ID *i* mod N every round *i*.

If a node which will become a cluster-head in a current round does not exit, the other nodes do not hear cluster-head announcement. In this case, all nodes rerun cluster-head selection after $T_{cluste}$, which is the time for broadcasting a cluster-head announcement to all nodes in the network considering propagation delay.

## 4.3 Routing-tree construction

CIPRA adopts a topology of a doughnut form based on a tree structure, in which the cluster head is located at the center. During a data-routing tree construction phase, each node selects a parent node which receives and fuses data transmitted from its child nodes.

During a clustering phase, non-cluster-nodes compute their tree level value (TLV) according to the distance from a cluster-head. TLV is a discrete number.

We note that the distance between tree levels, called a communication radius (*ComRadius*), is decided a priori based on node density in a monitored field considering communication connection among nodes.

In [7], the critical transmitting range for multi-hop connectivity was presented. The authors assume that the nodes are uniformly distributed in the network field and that each cell of size c × c in the network contains at least one node. In this case, the network is guaranteed to be connected if the transmission range Rt=(1+ $\sqrt{5}$ )c [7]. According to [7], we obtain that *ComRaius* is equal or longer than (1+ $\sqrt{5}$ )* ( $\sqrt{\frac{L \times L}{N}}$ ) where L is the length of a side of L × L field.

We can calculate the maximum tree level value ($TLV_{max}$) according to the longest length of a field and *ComRadius* using the following function.

$$TLV_{\max} = \frac{The\ longest\ length\ of\ a\ field}{ComRadius} + 1$$

The routing setup phase requires a number of iterations (steps), which we refer to as $N_{itr}$. Every step takes $T_R$, which should be long enough for the same tree-level nodes to transmit messages and for neighbors to receive messages. First, the routing setup phase starts from nodes that have the smallest TLV, 1, and terminates at the nodes that have $TLV_{max}$ （＝$T_{Nitr}$） in Figure 2.

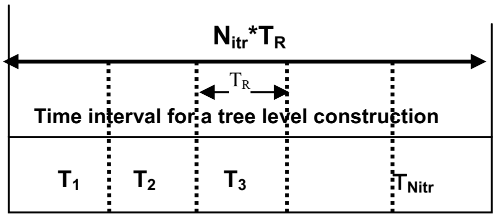


Figure 2: The time for routing-tree construction

Each non-cluster-head node computes its TLV as follows:

$$TLV = \left\lceil \frac{D_{CH}}{ComRadius} \right\rceil + 1$$

where $D_{CH}$ is the distance to its cluster-head based on the power of a signal received from the cluster-head.

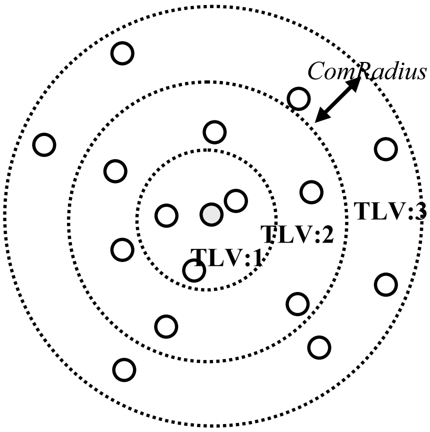


Figure 3: The TLV structure

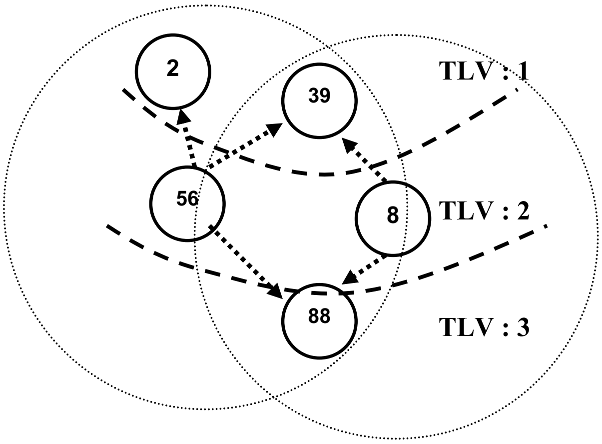


Figure 4: Broadcast for routing-tree construction

During iteration *i*, $i \leq N_{itr}$, nodes whose TLV is *i*, broadcast, using CSMA, a message which is composed of transients' parent node ID.

Nodes whose TLV is i+1 receive the message and decide a node that transmitted a signal of the message with the strongest power in the nodes that sent the message, as their parent node. When nodes of which TLV is i-1 receive the message, they list their child nodes and neighbor's child nodes. Then they turn off their radio after this iteration is finished. Nodes of which TLV is 1 elect the cluster-head as their parent node.

For example in figure 4, during iteration 2, node56 broadcasts a message:< parent node=node2> to neighbors such as node2, node 39 and node88. Node8 also broadcasts a message:<parent node=node39> to neighbors such as node39 and node88. Node88 of which TLV is 3 selects node8 as its parent node because node8 is closer to node88 than node56. Node2 lists node56 as its child node. Node39 also lists node8 as its child node. Then node2 and node39 turn off their radio.

The pseudo-code for each node is presented in Figure 5. If a node has never received a routing message the node selects its cluster-head as its parent node.

```
--------------------------------------------------------------------
Routing-tree Construction for each node $ID_i$
I. Initialize
    1. Compute its TLV
           $ID_i$ calculates the distance ($D_{CH}$) to the cluster-head
            based on the power received from the cluster-head
           $ID_i$ computes TLV by the following equation:
           $TLV = \frac{D_{CH}}{ComRadius} + 1$
    2. Setup Time
    3. Data transmission time (from 1 to $N_{itr}$)

II. Routing-tree Construction in Time Interval $T_i$ (i=1..$N_{itr}$)
    If TLV=1 then
        $ID_i$ broadcasts Route_Message [$ID_i$]
     Else
        $ID_i$ broadcasts Route_Message
                        [$ID_i$, $ID_i$'s parent   node ID]
     Else If  TLV = i-1 then
        $ID_i$ receives Route_Message from neighbors
        $ID_i$ lists its child nodes and
                  the child nodes of other nodes
        $ID_i$  turns off its Radio
     Else if  TLV = i+1 then
        $ID_i$ selects a node of TLV=i that transmitted a signal
        of the message with the strongest power in the nodes
        that sent the message, as its  parent node.
     Else
        $ID_i$  turns off  its Radio

III. No message comes to $ID_i$
 1.  If $ID_i$  does not   receive any message, it selects a
     cluster-head as its parent node.
--------------------------------------------------------------------
```

Figure 5: The pseudo-code for each node

During the routing set phase, each node consumes an amount of energy required to receive messages from several tentative parent nodes. To reduce reception-energy, CIPRA limits the number of messages received from several tentative parent nodes.

### 4.4 Data transmission

In this phase, we describe data flow over a data–routing tree to the base station via a single cluster head. After the routing setup phase terminates, each node knows its child nodes, its parent node and its TLV. During a data transmission phase, roughly speaking, each sensor waits for data from all its child sensors before sending up its aggregation to its parent node. After non-cluster-head nodes had sent data to their parent node, they turn off their radio.

The data transmission phase needs for a number of intervals, each of which is called a communication slot. The number of communication slots is equivalent to $TLV_{max}$ to transfer data to the cluster-head. In each interval, $T_{cs}$, transmitting and receiving data are done between hops. In the *i-th Tcs*, nodes whose TLV is i transfer data to their parent nodes and nodes whose TLV is $i$-1 aggregate data.

During a communication slot, the sensor delivers its packet to its parent. The parent node aggregates the transmitted data from its child sensors and transmits data to its parent node. These processes are repeated until the data reach the cluster-head. Cluster-head aggregates the gathered data and transfer it to the base station. Non-cluster-head nodes of which TLV is $TLV_{max}$, start to transfer data.

For instance, as shown in figure 6, while node 1 and node 2 transmit their data, node 11 waits for receiving them. After receiving the data, node 11 aggregates and transfers the data. The slot mechanism gives energy consumption advantage. To preserve sensor energy, sensors are put into sleeping mode. This is done during idle times after each node has finished sending its data.

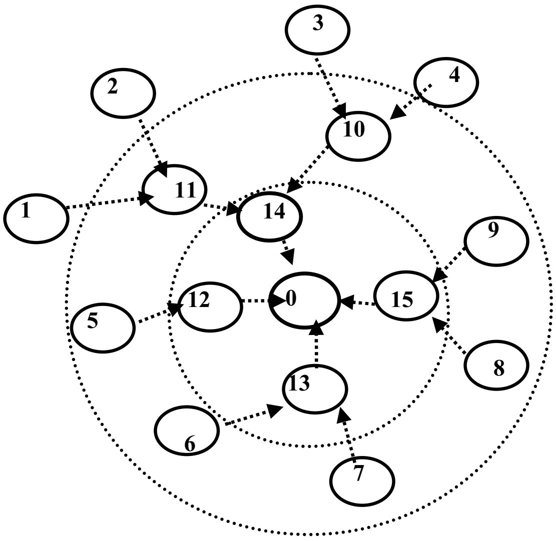


Figure 6: Data transmission

## 4.5 Topology change

As time goes by, nodes will exhaust their energy and dead nodes will increase over the network. The problems of communication breakdown will occur. To keep making communication connected between sensor nodes, CIPRA proposes several tips for fault tolerance.

First, each node consumes different energy according to the difference of a role such as a non-cluster-head or a cluster-head. If a non-cluster-head node holds energy only for routing setup and can not transfer data to its parent node during a data transmission phase, it should not participate in the routing setup phase to save its energy and reduce un-useful message loads. If a node does not hold energy required for taking a role of a cluster-head, it abandons to become the cluster-head. At that time the cluster-head selection is re-initiated. Also, a node which does not become a cluster member and does not have sufficient energy required to receive and transmit a data in a round is regarded as a dead node even though its energy does not wholly become exhausted. During the routing setup phase, the node broadcasts a network message : <parent node, parent node's TLV>. If a node receives the network-messages and has a less TLV than that of transients, it does not turn off its radio and waits to receive a routing message from other nodes which have one more TLV than TLV of transients. If the node which received network-messages has a greater TLV than that of transients, it checks other received messages. And then, if the node does not receive any routing messages, it transmits routing messages to TLV of the parent node of transients in order to let the parent node of transients know its existence as a child node.

Second, against isolated nodes which can not receive a message from other nodes, CIPRA extends a radio range of a node to maintain the network. In other words, each node perceives its neighbor nodes by receiving network messages or a process of neighbor discovery. If nodes recognize that the number of neighbor nodes is below a threshold which standardizes the number of live neighbors within a radio range of a previous round. The node that becomes a cluster-head broadcasts a massage which allows other all members to change their *ComRadius* by returning acknowledgements to a cluster-head announcement massage. All members change their *ComRadius*, compute their TLV and extend their radio range according to new *ComRadius*. Nodes periodically perform a process of neighbor discovery. However the neighbor discovery is not necessary every time because of a stationary network.

Last, the isolated nodes which can not receive a message from other nodes during its routing set phase, it select its cluster-head as its parent node.

## 5 EVALUATION

In this section, we evaluate the performance of CIPRA with a simulator. The aim of the sensor networks is to transmit data of a monitored environment to the base station with prolonged lifetime. Namely, if sensor nodes can't cover the monitored environment, data quality of a system will rapidly decrease. We therefore pay attention to a fact that the sufficient number of sensors must be kept up in a monitored field to maintain high data quality. We examine the network lifetime that is defined as the number of rounds until the energy of the first sensor node is run out.

We assume that 100 nodes are uniformly deployed in a monitored field with 100*100 $m^2$. Each node is uniformly and randomly located in a rectangular field of 10*10 $m^2$. The position of the base station is located at far away 75 from the closet point over the network. Further we assume each node aggregates data transmitted from its child nodes into a single packet whose size is 800 bits. The initial energy of each node is 0.25J. The LEACH protocol uses five clusters for a network composed of 100 nodes according to its optimal cluster-head selection equation [4-5].

## 5.1 Performance evaluation

### 5.1.1 A bound of ComRadius

First, we must evaluate a bound of *ComRadius* shown in Figure7. Suppose, under the assumption of each cell of size c×c in the network contains at least one node, *ComRadius* should be longer than or equal to $c \times \sqrt{5}$ that is a diagonal length of each cell. If *Comradius* is shorter than the bound length, there may not be any node in a TLV.

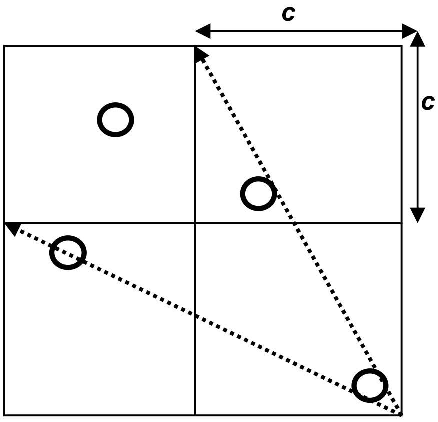
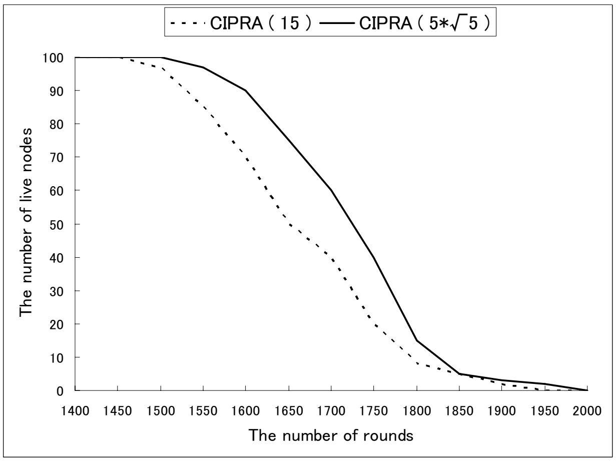


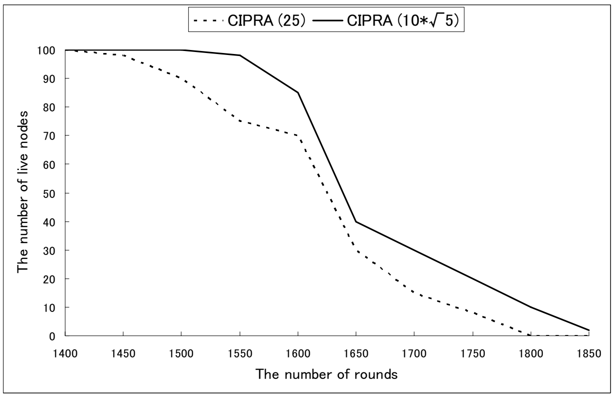
Figure 7 : The minimum distance (c×c)

### 5.1.2 The node density of a different field size

We evaluate the lifetime according to the different field size, 50×50($m^2$) and 100×100($m^2$). In Figure 8 (a) , the case of Field size 50*50 presents lifetime according to two different ComRadius: 5* $\sqrt{5}$ and 15, where 5* $\sqrt{5}$ is the minimum *Comradius*. *ComRadius* of 5* $\sqrt{5}$ holds longer lifetime than that of *ComRadius* of 15 because of less energy required to transmit and receive data. In Figure 8(b), nodes are located in the field whose size is 100*100. In both cases, the shorter the Comradius is, the longer the network lifetime is.



(a)Field size 50×50 : (25,175)



(b)Field size 100×100 : (50,175)

Figure 8: The network lifetime (number of rounds) according to two different filed sizes.

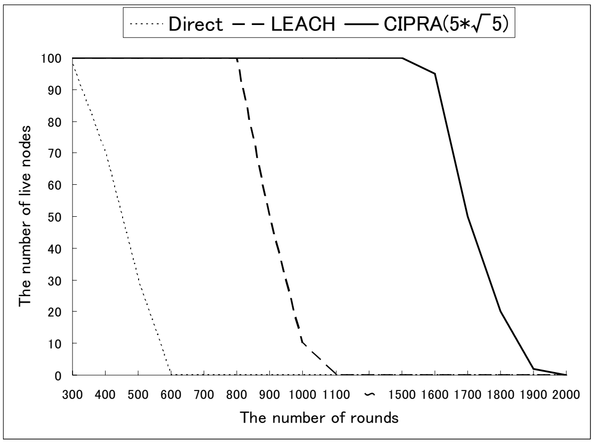
## 5.2 Comparison with LEACH and Direct transmission

We compare CIPRA with LEACH and Direct transmission. For LEACH, we specified that 5% of the nodes would be elected as cluster heads.

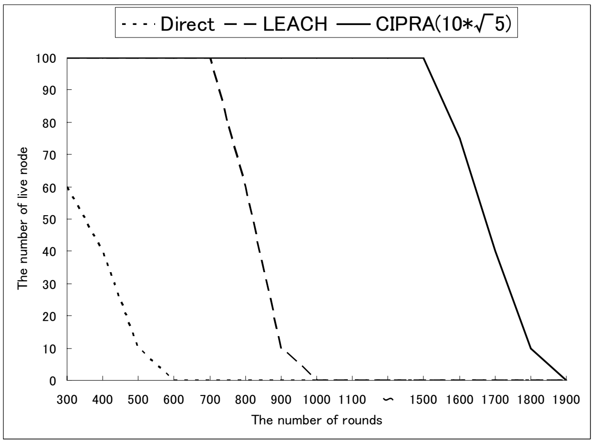
As shown in Figure 9, CIPRA has good improvement about the first death node compared with LEACH. Moreover the lifetime of the last node death of radio range $c \times \sqrt{5}$ of CIPRA is more than two-times longer than that of LEACH.

In LEACH, each member sensor transmits its data to its cluster head, and the cluster head only aggregates data. On the other hand, in CIPRA each node aggregates data using multi-hop communication across a tree path so that the amount of data is reduced. Therefore, when comparing with

LEACH, the transmission distance of each sensor in CIPRA is shorter than that in LEACH because each sensor in LEACH directly transmits data to a cluster head by single hop.



(a) Field size 50×50 : (25,175)



(b) Field size 100×100 : (50,175)

Figure 9: The number of rounds according to two different filed sizes: ComRadius of CIPRA is $c \times \sqrt{5}$ where c is 5 in (a) and 10 in (b)

## 6 CONCLUSION

In this paper, we discussed energy-aware and efficient data gathering mechanism, CIPRA, which is based on clustering and a data-routing tree using data aggregation at nodes in a cluster. CIPRA reduce energy required to construct a multi-hop routing-tree and transmit data by using hybrid of TDMA and CSMA and by shortening communication distance. CIPRA copes with changes over network (dead nodes) and maintains communication connect between sensor nodes.

Experimental results showed that our data gathering mechanism outperforms the direct scheme protocol and the LEACH protocol from the point of view of the network lifetime. Where not only the time of the first node death, but also the lifetime of the last node became prolonged.

## 7 FUTURE WORK

CIPRA reduces the energy required for transmitting data to the base station when selecting one cluster head. However when a network is composed of several hundreds or thousands sensors, it requires multiple cluster heads. In the cases, we will need an algorithm selecting multiple cluster heads considering the residual energy of each node. Furthermore the increases of the death nodes form holes which break off the communication between nodes. To deal with this, we have to adaptively increase the radio range according to the number of the death nodes in the radio range of each sensor.

## REFERENCES

[1] Ian F. Akyidiz, Weilian Su, Yogesh Sankarasuramniam, "A survey on sensor network", IEEE Communication Magazine, Vol 40, NO.8, pp.102-114, Aug 2002

[2] S. Bandyopadhyay, E. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor network", IEEE Infocom, San Francisco, CA, 2003

[3] Mohamed Younis, Moustafa Youssef, Khaled Arisha, "Energy-Aware routing in Cluster-based sensor networks", Proceedings of the 10th IEEE/ACM International Symposium on Modeling, Analysis, Simulation of Computer and Telecommunication System, Fort Worth, Octorber, 2002

[4] W. R. Heinzelman, A. Chandrakasan, H.Balakrishnan, "Application-sepcific protocol architecture for wireless microsensor sensor networks", IEEE transaction on Wireless Communication 1(4), pp. 600-670, 2002

[5] W. R. Heinzelman, A. Chandrakasan, H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks", in 33rd Annual Hawaii International Conference on System Sciences, pp.3005-3014 , 2000

[6] Stemphanie Lindsey, and Cauligi S. Ranhavendrea, "PEGASIS: Power-Efficient Gathering in Sensor Information Systems", ICC 2001

[7] O. Younis and S. Fahmy, "Distributed clustering in ad-hoc sensor networks: A hybrid, Energy-efficient approach", In Proceeding of IEEE INFOCOM, August 2004

[8] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks", In Proc,

6th Annual International Conference on MobiCom 2002. pp. 56-67, Boston, ACM press Augest 2000

[9] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, "TAG: a tiny aggregation service for ad-hoc sensor networks", In Pro. of ACM WSNA'02, 2002

[10] Jonathan Beaver, Mohamed A. Sharaf, Alexandros Labrinidis, and Panos K. Chrysantihis, "Location-aware Routing for Data Aggregation in Sensor networks" In: Proceedings of Geo **Sensor Networks** Workshop 2003

[11] Vivek Mhatre, Catherine Rosenberg, "Design guidelines for wireless sensor networks: communication, clustering and aggregation" to appear in Ad Hoc. Networks Journal 2003.

[12] H. Chan and A. Perrig, "ACE: An Emergent Algorithm for Highly Uniform Cluster Formation", Proc. First European Workshop Sensor Networks, Jan. 2004

[13] M, Bhardwaj, A. Chandrakasan, and T. Garnett, "Upper bounds on the lifetime of sensor networks", in IEEE International Conference on communications, pp.5-67, 2001

[14] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, "Tinyos: An operating system for wireless sensor networks" Springer-Verlag, To Appear. 2004.

[15] L. M. S. C. of the IEEE Computer society, "Wireless lan medium access control (mac) and physical layer (phy) spec-ification," in IEEE Std 802.11, 1997.

[16] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems. New York, NY, USA: ACM Press, pp. 138–149, 2003