RCCT: Robust Clustering with Cooperative Transmission for Energy Efficient Wireless Sensor Networks

Mahbod Ghelichi
Electrical engineering department
Sharif university of technology
Tehran, Iran
+98-935-2160794
m_ghelichi@ee.sharif.edu

Seied Kazem jahanbakhsh
Electrical engineering department
Sharif university of technology
Tehran, Iran
+98-912-2035105
jahanbakhsh@ee.sharif.edu

Esmaeil Sanaei
Electrical engineering department
Sharif university of technology
Tehran, Iran
+98-912-1127471
sanaei@sharif.edu

ABSTRACT

Data gathering is a common but critical operation in many applications of wireless sensor networks. Innovative techniques that improve energy efficiency to prolong the network lifetime are highly required. Clustering is an effective topology control approach in wireless sensor networks, which can increase scalability and lifetime. Collaboration of hundreds or thousands of cheap micro sensor nodes allows users to accurately monitor a remote environment by intelligently combining the data from the individual nodes. These networks require robust wireless communication protocols that are energy efficient and provide low latency. In this paper, we develop and analyze an efficient cooperative transmission protocol with robust clustering (RCCT) for sensor networks that considers a fault-tolerant and energy-efficient distributed clustering with minimum overhead. RCCT distributes energy load by energy-aware member selection for cooperative data transmission. Simulation results show a better performance of RCCT as compared to the conventional protocols.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design – *distributed networks*, *network topology, network communication, wireless communication.*

General Terms

Algorithms, Performance, Design, Reliability.

Keywords

Clustering, cooperative transmission, fault tolerant algorithm, wireless sensor network.

1. INTRODUCTION

Rapid technological advances in MEMS and wireless communication have enabled the deployment of large scale wireless sensor networks. The potential applications of sensor networks are highly varied, such as environmental monitoring, target tracking, and battlefield surveillance [1, 5]. Distinguished from traditional wireless networks, sensor networks are characterized by severe power, computation, and memory constraints. Due to the strict energy constraint, energy resource of sensor networks should be managed wisely to extend the lifetime of sensors.

The behavior of such networks can be highly unpredictable because of the operating characteristics of the nodes and the randomness in which the network is set up. Hence the devised algorithms should consider failure of a network as a rule rather than as an exception, and can handle this more efficiently.

In order to achieve high energy efficiency and increase the network scalability, sensor nodes can be organized into clusters. The high density of the network may lead to multiple adjacent sensors generating redundant sensed data, thus data aggregation can be used to eliminate the data redundancy and reduce the communication load [4]. In periodical data gathering applications, both methods promise to efficiently organize the network since data collection and processing can be done “in place”.

Many possible situations can be defined for these micro sensor networks. Our assumptions are as follows:

- All nodes in the network are homogeneous and energy constrained.
- The base station is immobile and all nodes are able to reach the base station.
- The propagation channel is symmetric.
- Nodes can use power control to vary the amount of transmission power.

In this paper, we propose and evaluate RCCT (Robust Clustering with Cooperative Transmission), a LEACH-like clustering scheme, that balances the communication load by means of cooperative transmission. We also suggest *imperceptible monitoring fail-safe* which guarantees fault-tolerant behavior of the protocol with the least overhead. LEACH is a stochastic cluster head selection algorithm, which divides the operation of the network into rounds, each of which consists of a set-up and a steady-state phase. During steady-state phase data transfer is done through TDMA scheme for communicating with cluster head and CDMA scheme for avoiding collision between adjacent clusters.

RCCT partitions the network into certain number of non-overlapping clusters with one cluster head almost in the middle of each one. Cluster heads perform load balancing strategy through energy aware transmission. Load balancing will cause cluster heads not to lose their power quickly, leading to longer steady-

MOBILWARE 2008, February 13-15, Innsbruck, Austria

DOI 10.4108/ICST.MOBILWARE2008.2870

state phases that amortize the overhead of successive set-up periods. As the operation of a cluster is fully dependent on its head, a fault-tolerant frame-work is introduced by reliable monitoring of cluster head health. As will be shown, this health monitoring is done efficiently by all of the cluster members in the background of network operation. RCCT is fully distributed, low-overhead and more energy efficient and simulation results show that it outperform LEACH and other classical clustering algorithms.

The reminder of this paper is organized as follows: Section II covers related work in this area; Section III describes the protocol; Section IV presents the simulation results, and section V concludes the paper.

2. RELATED WORKS

Using clustering enables better resource allocation and helps improve power control. Therefore, many clustering algorithms have been developed for wireless sensor networks in recent years. In LEACH [10], Heinzelman et al. used randomized rotation of the cluster heads to achieve load balancing and power efficiency but being fully dependent on random actions incurs it a lot of deficiency. Here, each node has a certain probability of becoming a cluster head per round, and the task of being a cluster head is to send aggregated data packets to the base station by single hop. In order to avoid contention and collision a hybrid TDMA-CDMA MAC protocol has been proposed for low-energy operation. Using a TDMA approach saves energy by allowing the nodes to remain in the sleep state, with radios powered-down, for a long time. PEGASIS [9] improves the performance of LEACH on lifetime and energy conservation by connecting the sensors into a chain. But delay is significant although the energy is saved. HEED [8], extends LEACH by incorporating communication range limits and intra-cluster communication cost information, where the cluster head selection is carried out periodically according to a hybrid of the node residual energy and node proximity to its neighbors through constant time iterations. Both HEED and LEACH require re-clustering after a period of time to achieve load balancing and time synchronization, causing extra energy consumption.

Existence of mobile nodes in the network besides fault tolerant behavior of clusters has been investigated in [2] where an organizer node is assigned to each cluster for handling mobile nodes and monitoring the operation of cluster head actively. This approach can not satisfy power efficiency and lifetime maximization because it doubles always-awake nodes in each cluster (organizer and cluster head). Furthermore, none of the protocols described above take into account the terrain dynamics and time varying channel condition, which can be exploited to avoid fading and packet loss. For this end, cooperative communication has also been studied in recent years. In [3] three cooperative methods, namely detect and forward, amplify and forward, and coded cooperation are presented. RCCT builds on these works by creating a new ad-hoc cluster based algorithm that better suits wireless sensor network applications.

In this paper we developed a robust, reliable and low-overhead fully distributed clustering hierarchy which strictly considers power efficiency and load balancing. In the set-up phase, the cluster head is elected by semi-probabilistic localized competition, which is unlike LEACH, and with no iteration, which differs from HEED. The large variance of the number of clusters which LEACH wants to fix besides unbalanced communication load over cluster heads and lack of reliability on cluster heads failure condition, are the most important issues we want to tackle in our protocol.

3. PROTOCOL DESIGN

3.1 Network Architecture

In this protocol, our motivation is to meet the unique requirement of data gathering in wireless sensor networks for which LEACH is proposed, i.e. single-hop communication. In this typical data gathering application, individual node's data are often correlated. Therefore, periodically sensed data are combined by means of data aggregation techniques and then transmitted to the base station (BS). Base station as an end user, analyzes the data to draw some conclusions about the activities in the field.

For the development of RCCT, we adopt a few reasonable assumptions for simplicity. For the network it is assumed that N sensors are uniformly dispersed within a square field A. All nodes are able to use continuous power control and transmit with enough power to reach the BS if needed. All sensors are homogeneous and location-unaware. For better simulation and comparison with previous works, it also assumed that nodes always have data to send to the BS.

In RCCT, lifetime of the network is split into fixed-time rounds. As shown in Figure 1, each round starts with a set-up phase followed by a steady-state phase. Topology structure is formed into clusters during the set-up phase and some nodes compete to become cluster head in this fixed period of time, which itself is divided into three sub-period. In the steady-state phase, nodes are scheduled in a variable-length TDMA scheme to avoid collision and better management. There are two different kinds of time slots in our approach: One for common operation of data transfer from cluster members to aggregation point (cluster head) and the other for cooperative transmission which will be explored in details in the following sections.

3.2 Cluster Formation Algorithm

The title RCCT forms clusters by using a distributed algorithm, where nodes make semi-autonomous decisions without any centralized control. Our objective is to obtain a cluster formation algorithm such that there are a certain number of clusters, k, during each round. In addition, we try to build equal size clusters to exploit power control readily and efficiently. At the first part of every set-up phase, each node decides whether to participate in cluster head competition or not. The decision is taken based on the suggested number of cluster heads for the network (specified a priori) and the number of times the node has been a cluster head so far. Each node chooses a random number between 0 and 1. If the selected number is less than a threshold $T(n)$, then, the node enters the competition. The threshold $T(n)$ is set as follows [10]:

$$T(n) = \begin{cases} \dfrac{K}{N - K * (r \mod \frac{N}{K})} & if \quad n \in G \\ 0 & otherwise \end{cases} \tag{1}$$

where N is the total number of nodes, K is the desired number of cluster head nodes, r is the number of current round and G is the set of nodes that have not been a cluster head in the last N/K rounds. Therefore, every node will be a cluster head for a while during the last N/K rounds. For $r = 0$, we have $T(n) = \frac{K}{N}$ and for

$r = \frac{N}{K} - 1$, $T(n)$ will be 1. In other words, as illustrated in lines 4-6 of Figure 2, if a node has not been chosen as a cluster head recently, it would have been a candidate for the reminder rounds.

Cluster head selection algorithm in LEACH was purely based on the random function described in equation (1) without considering the energy and location of cluster head nodes. This approach, conducted statically in a two-step setup phase, would result in the constitution of some overlapping and asymmetric clusters unfairly distributed across the network. In other words, all the nodes with the chosen random number greater than $T(n)$ would be cluster head without further exploration. RCCT protocol, instead, have added an additional correction period in a dynamic three-step setup phase considering distance between competitor nodes to select suitable and uniformly distributed cluster heads. All of these considerations help RCCT to achieve better energy consumption.

In RCCT cluster formation algorithm, competitor nodes would try to broadcast an *invitation message* (INV) using a non-persistent carrier-sense multiple access (CSMA) Mac protocol. This message is a small message containing the node's ID and a header that distinguishes this message as an *invitation message*. However, this message must be broadcast powerfully to reach all of the nodes in the network. These powerful invitation messages ensure the elimination of collisions when CSMA is used, since there is no hidden terminal problem. All the nodes in the network can calculate their relative distance to the inviters by means of received signal strength and compare them with some distance threshold in order to make decision about which cluster to join or what action to do in the next parts of setup phase. This distance threshold is specified according to the area of the network and the ideal number of cluster heads[1].

During the first part of the set-up phase, if a competitor node finds itself closer than the distance threshold to an inviter, then it gives up competition and waits for the *joint-time* to choose its ideal cluster head. In other words, when two or more competitor nodes are situated near each other within a specified distance threshold, all of them will certainly give up competition in favor of the node which broadcasts its invitation message first with respect to CSMA channel access. In the second period of the set-up phase, if a common node, which has not been cluster head in a few recent rounds, couldn't find an inviter closer than the threshold, would claim to be cluster head and broadcast an *invitation message* just like the first step. In this approach, our goal is that if a node becomes a cluster head at the end of the second period, there will not be another cluster head within its cluster diameter. It will be shown in the section IV that this strategy partitions the network into certain number of non-overlapping clusters.

In the third part of the set-up phase, each common node selects its cluster head based on the received *invitation message* power. Non-cluster head nodes respond to their selected cluster head through a *join-REQ* message which also piggybacks the nodes power level information. From now on, cluster head and its members have enough information to exploit efficient power control in their cluster. The cluster heads act as local control centers to coordinate the data transmissions in their cluster. Therefore, the cluster head node sets up a TDMA schedule in addition to a unique spreading code and transmits them to the nodes in the cluster. All the nodes in the cluster transmit their data to the cluster head just in its corresponding time slot, using this spreading code. This is known as transmitter-based code assignment [6], since all transmitters within the cluster use the same code. These considerations ensure that there are no collisions among data messages and also allows the radio components of each member node to be turned off at all times except during pre-specified time slots. Figure 2 expresses the overall cluster formation mechanism in a simplified pseudo code.

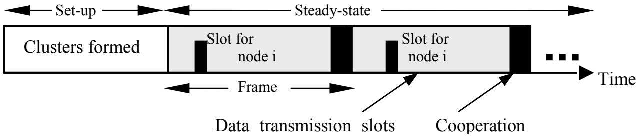

Figure 1. Time line showing RCCT operation.

Algorithm 1: Cluster formation

```
1:  First period:
2:  μ←RAND(0,1)
3:  T←T(n)                                   // given by eq. (1)
4:  if HasnotBeenCH = TRUE   then
5:      if μ < T  then
6:          beCompetitor ← TRUE
7:      end if
8:  end if
9:  if beCompetitor = TRUE  then
10:     TrytoClaimCH (ID)
11: else
12:     listen and wait for the second period
13: end if
14:     On receiving invitation message from a CH:
15: if distance < threshold  then
16:     if beCompetitor = TRUE  then
17:         give up CH claiming
18:     end if
19:     Invited ← TRUE
20:     go to join-REQ
21: end if
22: Second period:
23: if Invited = FALSE  then
24:     TrytoClaimCH(ID)
25: end if
```

Figure 2. Cluster formation pseudo code.

3.3 Steady-state Operation

After the cluster deployment, the steady-state phase begins and the network starts its normal operation. The steady-state operation is broken into frames. These frames themselves include two kinds of slots, namely *data transmission slots* and *cooperation slots*. During this phase, member nodes are asleep all the time during *data transmission slots* except to their own time slot in which they transmit data to the cluster head. These data contain node ID, the measure of sensed parameter and residual energy of the node. The information about residual energy of each node will be exploited in future for load-balanced routing. The duration of each slot where a node transmits data is constant, so the data transmission slot time depends on the number of nodes in the cluster.

---

[1] Calculation of optimal number of cluster head is available in [10].

```
Algorithm 2: Steady-state operations
while (transmission data slot time) do
    if IsCH = TRUE then
        receive data from members;
    else
        sleep;
        sense the parameter;
        if IsYourTimeSlot = TRUE then
            wake up;
            TransmitToCH (ID, $E_{residual}$, Data);
        end if
    end if
end while
while (cooperation slots time) do
    wake up;
    if IsCH = TRUE then
        aggregate received data;
        select the most energetic node;
        transmit back aggregated data;
    else
        receive backed aggregated data;
        if YouAreSelected = TRUE then
            transmit to BS;
        else
            sleep;
        end if
    end if
end while
```

Figure 3. Steady-state operation pseudo code.

Cooperation slots are consisting of a few time slots located in the end of each frame, just after *data transmission slots*. During *cooperation slots*, the cluster head carries out data aggregation and broadcasts the aggregated data back to the member nodes which are all awake now. The most energetic node then transmits the aggregated data to the BS, where other nodes come back to sleep state to save energy. We note that some overhead may be incurred by broadcasting the data back to member nodes. However, since this broadcast occurs only over a short distance within a cluster, the overhead is negligible towards the outstanding benefits it presents. We elaborate further on this issue in the next part. Figure 3 illustrates the overall operation of steady-state phase in a simplified pseudo code.

3.4 Load Distribution and Fault-tolerance

Evenly distribution of the energy load among all the nodes in the network is desirable, so that there are no overly-utilized nodes that will run out of energy before the others.

As in the traditional algorithm, being a cluster head node is much more energy intensive than being a non-cluster head node, this requires that each node takes its turn as cluster head. Re-choosing cluster head naturally incurs a lot of overhead as we encounter it in our set-up phase. Our objective is to balance the load among the sensors, within a cluster. For achieving this goal, we introduce a cooperative approach in which cluster members relay aggregated data while cluster head aggregates them. As discussed previously, member nodes transmit data packets to cluster head during *data transmission slots*. Residual energy of each node can be simply inserted in the header of the data packets. Therefore, cluster heads are wisely able to choose a member with the highest residual energy to transmit aggregated data back to it. The chosen node will take the responsibility for long distance and energy intensive transmission of aggregated data to BS. This high-power node selection is performed in each frame during *cooperation slots*. With this heuristic mechanism, fine grained load balancing is conducted locally in each cluster under the supervision of the cluster head only. This approach not only equalizes the energy consumption of member nodes, but also unburdens cluster heads from one of the most energy intensive responsibilities. Interestingly, such a load balancing heuristic and cluster head unburdening will also improve lifetime and overall energy consumption indirectly by amortizing the overhead of early re-clustering. In other words, we do not need to run the energy intensive set-up phase very early like other previous clustering algorithms, to avoid cluster head energy draining. In this way, control overhead caused by the set-up phase is decreased by an order of magnitude compared with LEACH algorithm.

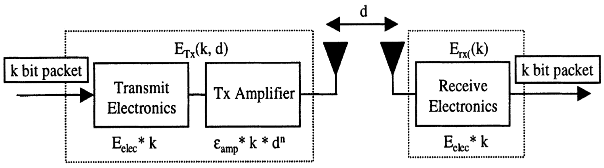

Figure 4. Radio energy dissipation model.

Cluster heads are the heart of each cluster. All the vital affairs like data aggregation and node selection for long haul transmission are managed by cluster heads. As a result, cluster head failure, which might occur because of energy draining or any other kinds of node problems, will lead to malfunctioning of the whole cluster. Unlike the previous protocols which needed re-clustering to overcome this problem, we propose *imperceptible monitoring fail-safe* to add robustness to our clustering scheme.

As mentioned earlier, all the member nodes are awake during *cooperation slots* to get the aggregated data transmitted back by the cluster head. If member nodes couldn't hear from their cluster head in two consecutive frames, they would run a local head selection algorithm. Local head selection algorithm is just like the set-up phase with some differences. For instance, the cluster spreading code is previously assigned; consequently, local set-up messages make no collision with adjacent clusters through using this code.

4. SIMULATION RESULTS

In this section, we evaluate the performance of RTTC protocol implemented with ns-2 [7]. As RCCT inherits its basics from LEACH, we adopt the same network and energy model for better comparison. For simulation a network with 100 nodes is used in which nodes are randomly distributed between $(x = 0,\ y = 0)$ and $(x = 100,\ y = 100)$ with the BS at location $(x = 100,\ y = 175)$. The bandwidth of the channel was set to 1 Mbps, each data message was 500 bytes long, and the packet header for each type of packet was 30 bytes long.

We assume a simple model for the radio hardware energy dissipation where the transmitter dissipates energy to run the radio electronics and the power amplifier, and the receiver dissipates energy to run the radio electronics, as shown in Figure 4. For the experiments described here, both the free space (d^2 power loss) and the multi-path fading (d^4 power loss) channel models were

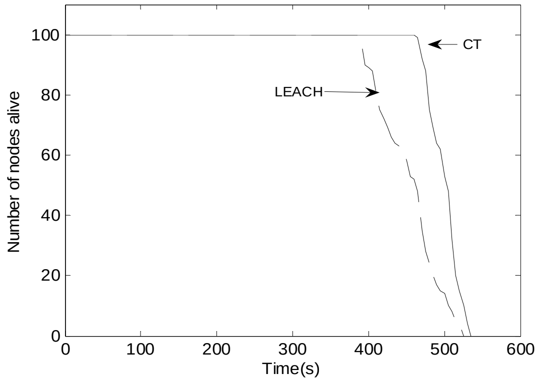

Figure 5. Number of nodes alive over time in CT comparing to LEACH.

used, depending on the distance between the transmitter and receiver [10]. Power control can be used to invert this loss by appropriately setting the power amplifier—if the distance is less than a threshold, the free space (fs) model is used; otherwise, the multi-path (mp) model is used. Thus, to transmit an ***l***-bit message to a distance *d*, the radio expends:

$$E_{Tx}(l,d) = E_{Tx-elec}(l) + E_{Tx-amp}(l,d) = \begin{cases} lE_{elec} + l\epsilon_{fs}d^2, & d < d_0 \\ lE_{elec} + l\epsilon_{mp}d^4, & d < d_0 \end{cases} \quad (2)$$

and to receive this message, the radio expends:

$$E_{Rx}(l) = E_{Rx-elec}(l) = lE_{elec} \quad (3)$$

The electronics energy, E_{elec}, depends on factors such as the digital coding, modulation, filtering, and spreading of the signal, whereas the amplifier energy, $\epsilon_{fs}d^2$ or $\epsilon_{mp}d^4$, depends on the distance to the receiver and the acceptable bit-error rate. For the experiments described in this paper, the communication energy parameters are given in Table 1.

Table 1. Simulation parameters

Parameter	Value
Network coverage	(0,0)~(100,100)
Base station location	(100, 175)m
Number of nodes	100
d_0	87 m
Initial energy	2 J
E_{elec}	50 nJ/bit
ε_{fs}	10 pJ/bit/m^2
ε_{mp}	0.0013 pJ/bit/m^4
E_{DA}	5 nJ/bit/signal

As shown in [10], the optimal number of cluster heads for this problem is around 5. We also chose the distance threshold for RCCT correction period (in the setup phase) heuristically equal to 1.5 times of ideal cluster radius. In determining distance threshold, one should consider the ideal number of clusters according to the area of the network and the area of each cluster in addition to the minimum acceptable distance between adjacent cluster heads. The selected distance threshold not only guarantees the number of cluster heads to lie near 5 but also avoids the existence of overlapping clusters properly. In this experiment, each node begins with only 2 J of energy and an unlimited amount of data to send. Each round also lasts for 60 s.

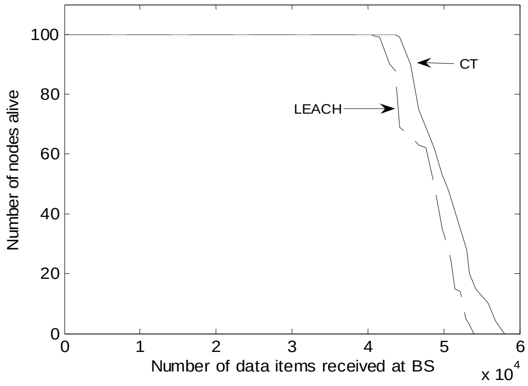

Figure 6. Total amount of data received at BS in CT comparing to LEACH.

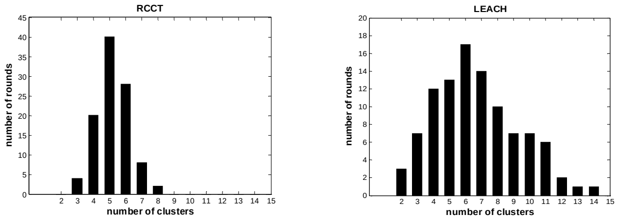

Figure 7. Distribution of the number of clusters.

We conducted our simulation in two separate parts for better understanding of each contribution. At first, we examined the energy efficiency and data transfer of RCCT in the absence of clustering contributions (correction parts of setup phase)[2] in figure 5 and 6. Figure 5 clearly shows the effect of load balancing mechanism resulting from cooperative transmission. Overall lifetime is not improved so much here, but the time between the death of first node and the last node is shorter than LEACH.

According to Figure 7, by adding correction part of setup phase for efficient clustering, the calculated value for average and variance of the number of clusters in RCCT (μ=5.2, δ=0.27) is more optimal and steadier than that in LEACH (μ=6.6, δ=0.44). In RCCT, nodes would claim being cluster head if they couldn't find a suitable one in the close area, thus the number of selected cluster heads won't be too small. On the other hand, candidate nodes simply give up cluster head claiming when they are in the range of another cluster head, thus the number of selected cluster head will neither be too large. In fact, the number of clusters using RCCT depends on the range and distance threshold of tentative cluster heads.

Finally, we examined the energy efficiency of perfect RCCT by investigating the network lifetime. Figure 8 and 9 show the number of sensor nodes still alive over the simulation time. RCCT

---

[2] This sliced protocol is shortly named CT.

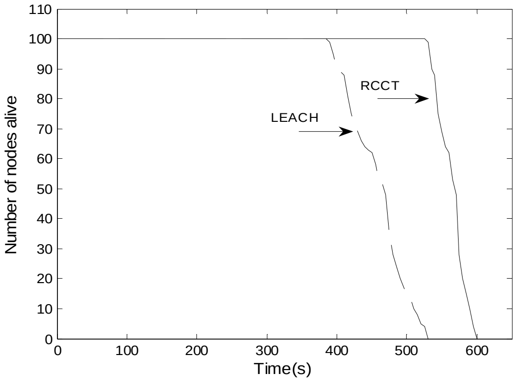

Figure 8. Number of nodes alive over time in RCCT comparing to LEACH.

clearly improves the network lifetime (both the time until the first node dies and the time until the last node dies) and amount of data delivered to base station over LEACH. Energy efficiency of RCCT is mostly related to the equal, symmetric and uniform clustering scheme. The small interval between the time until the first node dies and the time until the last node dies implies that RCCT has successfully balanced the communication load in the whole network.

5. CONCLUSION AND FUTURE WORKS

In this paper, we proposed a novel energy-efficient and fault-tolerant clustering mechanism applied for periodical data gathering. RCCT produces a uniform distribution of cluster heads across the network with the least overhead to exploit efficient power control. To address the problem of load balancing, we also introduced cooperative transmission scheme which not only balances the communication load but also amortizes the set-up control overhead with the objective of improving the network lifetime.

Network resiliency and robustness of clusters was also considered in *imperceptible monitoring fail-safe* with respect to the dynamic nature of the environment in which sensor networks are deployed. Simulation results indicate that the lifetime of the network is clearly extended with respect to RCCT also supports fault-tolerant behavior. In the world of cooperative communication and MIMO, different techniques like multi-user diversity, cooperative diversity and channel quality can be further analyzed in sensor networks.

Parameters of our mechanism, such as power threshold in cluster formation and the length of steady-state phase can be tuned to optimize energy preservation. We will try to find a solution that could determine the optimal value of these parameters according to network scale in our future work.

6. ACKNOWLEDGMENTS

The authors would like to express their sincere thanks to the Iran Telecommunication Research Center (ITRC) for supporting this work.

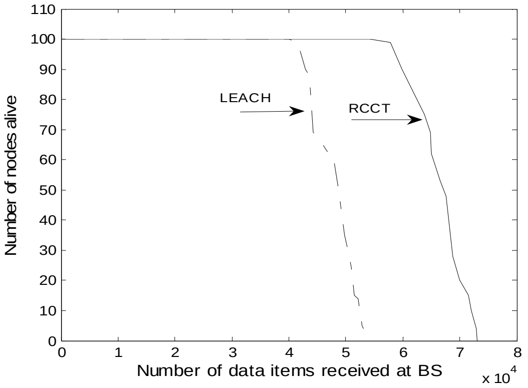

Figure 9. Total amount of data received at BS in RCCT comparing to LEACH.

7. REFERENCES

[1] A. Chandrakasan, R. Amirtharajah, S. Cho, J. Goodman, G. Konduri, J.Kulik, W. Rabiner, and A. Wang, " Design considerations for distributed microsensor systems, " in *Proc. IEEE Custom integrated circuits conf. (CICC)*, San Diego, CA, May 1999, pp. 279-286.

[2] A. Khadivi and M. Shiva, "FTPASC: A Fault Tolerant Power Aware Protocol with Static Clustering for Wireless Sensor Networks", *IEEE* International Conference on Wireless and Mobile Computing, pp. 397-401, WiMob 2006.

[3] A. Nosratinia, T. E. Hunter, and A. Hedayat, "Cooperative communications in wireless networks," *IEEE Commun. Mag.*, pp. 74–80, Oct. 2004.

[4] B. Krishnamachari, D. Estrin, and S. Wicker, "The Impact of Data aggregation in Wireless Sensor Networks", in *Proceedings of IEEE Int'l Conference on Distributed Computing Systems Workshops (ICDCSW)*, 2002, pp. 575-578.

[5] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey ensor Networks", *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102-114, 2002.

[6] L. Hu, "Distributed code assignment for CDMA packet radio networks", *IEEE/ACM Trans. networking*, vol. 1, pp. 668-677, Dec. 1996.

[7] Network Simulator, http://www.isi.edu/nsnam/ns

[8] O. Younis and S. Fahmy, "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks", *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 660-669, 2004.

[9] S. Lindsey, C. Raghavendra, K. M. Sivalingam, "Data Gathering Algorithms in Sensor Networks Using Energy Metrics", *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 9, pp. 924-935, 2002.

[10] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks", *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660-670, 2002