KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS VOL. 5, NO. 2, February 2011 247

A Reporting Interval Adaptive, Sensor Control Platform for Energy-saving Data Gathering in Wireless Sensor Networks

Wook Choi[1], Yong Lee[2] and Sangchul Kim[1]
[1] Department of Computer Science and Engineering, Hankuk University of Foreign Studies, S. Korea
[e-mail: wukichoi@gmail.com, kimsa@hufs.ac.kr]
[2] Department of Information and Communications Engineering, ChungJu National University, S. Korea
[e-mail: yleehyun@gmail.com]
*Corresponding author: Sangchul Kim

Received December 1, 2010; revised January 29, 2011; accepted February 24, 2011;
published February 28, 2011

Abstract

Due to the application-specific nature of wireless sensor networks, the sensitivity to such a requirement as data reporting interval varies according to the type of application. Such considerations require an application-specific, parameter tuning paradigm allowing us to maximize energy conservation prolonging the operational network lifetime. In this paper, we propose a reporting interval adaptive, sensor control platform for energy-saving data gathering in wireless sensor networks. The ultimate goal is to extend the network lifetime by providing sensors with high adaptability to application-dependent or time-varying, reporting interval requirements. The proposed sensor control platform is based upon a two phase clustering (TPC) scheme which constructs two types of links within each cluster – namely, *direct link* and *relay link*. The direct links are used for control and time-critical, sensed data forwarding while the relay links are used only for multi-hop data reporting. Sensors opportunistically use the energy-saving relay link depending on the user reporting, interval constraint. We present factors that should be considered in deciding the total number of relay links and how sensors are scheduled for sensed data forwarding within a cluster for a given reporting interval and link quality. Simulation and implementation studies demonstrate that the proposed sensor control platform can help individual sensors save a significant amount of energy in reporting data, particularly in dense sensor networks. Such saving can be realized by the adaptability of the sensor to the reporting interval requirements.

Keywords: Wireless sensor network, data gathering, clustering, adaptive sensor control

DOI: 10.3837/tiis.2011.02.001

1. Introduction

Wireless sensor networking is regarded as a promising technology in realizing the upcoming ubiquitous and context-aware computing, smart grid [1], and cyber physical space [2] environments. Wirelessly-connected sensors can serve as an infrastructure for providing fundamental information to predict and describe physical phenomena forming an autonomous wireless sensor network. Wireless sensor networks (WSN) can be deployed indoors and outdoors and also can be embedded in other systems, substituting for our sensory organs in inaccessible or inhospitable areas. Depending on the deployment platform, there are a variety of applications for such sensor networks, for example, environment or equipment monitoring, intrusion detection, smart home/smart space, surveillance [3][4][5], and space exploration, etc. The wireless sensor network can be characterized by a large number of nodes which operate under highly limited resource environments such as limited bandwidth, battery, computation power, and storage space. Sensors in such a network are equipped with data processing, wireless communication, sensing, and possibly mobilizing units. Due to the limited operating environment of sensors and the requirements for high density, then network scalability and reliability become critical for deploying WSN in a desired area to be seamlessly monitored.

Basically, sensor networks are task-specific where the tasks are given by the users for specific applications. In addition, the reporting frequency is determined by the application used [6]. Based upon the given task, sensors sense phenomena in their vicinity and transmit such sensed data over the network to the user who may update the sensors' behavior by sending a new task specification or data reporting interval, depending upon the analysis of the delivered data. The high network nodal density may lead to multiple sensors generating and transmitting redundant sensed data, which results in unnecessary power consumption significantly decreasing the network's lifetime. The sensors' actions, e.g., data transmission/reception and target sensing, result in a certain amount of energy consumption. Amongst them, the energy consumption for data communications is the most critical. Therefore, using the most energy-saving communication link or minimizing the number of communications by eliminating or aggregating redundant sensed data beforehand saves a large amount of energy, which ultimately leads to the network longevity. To achieve this goal, the network should be designed to seek local optimization driving optimal use of a limited resource such as energy collaborating with neighbors. For example, clustering [7][8], data aggregation (fusion) [9][10], and energy-aware load-balanced routing [11][12] can contribute to the local optimization. In this paper we focus on the use of clustering to design a sensor control platform for energy-saving data gathering.

1.1 Motivations

Such motivation is justified depending on the type of applications required. For example, achievable energy conservation can further be enhanced by meeting application-specific requirements such as the data reporting interval. Considering high data redundancy resulting from high node density, sensors need to be designed to eliminate data redundancy beforehand to conserve energy that would otherwise be consumed for unnecessary data transmissions. On the down side, such data redundancy elimination incurs some data forwarding delay due to in-network data processing. Some applications may not tolerate such delay to successfully complete a given task. Therefore, the reporting interval of a sensor needs to be flexibly controlled, yet maintaining the data redundancy elimination capability in order to maximize energy savings within the user/application-specific reporting interval requirements.

1.2 Our Contribution

In this paper, we propose a reporting interval, adaptive, sensor control platform for energy-saving data gathering in wireless sensor networks based on a clustering architecture. The ultimate goal is to extend the network lifetime by making sensors highly adaptive to the user-specified, time-varying, or application-specific reporting interval constraint. This is achieved through the topological connectivity management of a cluster for data forwarding. We characterize the proposed sensor control platform for energy-saving data gathering by addressing our contributions to the following:

- A two phase clustering (TPC) scheme is introduced as a basis for the proposed reporting interval supporting adaptive sensor control. The TPC divides the network into clusters and sets up two types of links within each cluster: *i*) *direct link* for control and time-critical data forwarding and *ii*) *relay link* for less critical data forwarding allowing for further energy-saving improvements.
- A reporting interval, adaptive, sensor control scheme is developed in which each sensor selects either the relay link or the direct link to transmit their sensed data after joining a cluster. The link selection depends on a forwarding pattern given by the cluster head that computes how many cluster members can use the relay link based on a given reporting interval constraint.
- We introduce a link quality-aware data forwarding scheduling scheme which gives a longer data forwarding opportunity to the nodes with a relatively lower link quality. Eventually, it helps to increase data packet delivery ratio reducing transmission delay.

Simulation study shows that our proposed platform can achieve an appreciable improvement on energy-saving in collecting data from the sensors. We present how to opportunistically control sensors to use the relay link and link quality-aware data forwarding scheduling. Furthermore, we implemented the proposed platform on a testbed [13] to more realistically evaluate the energy-saving capability in a real environment.

The remainder of this paper is organized as follows. Related work is presented in Section 2. Section 3 introduces basic notations, some definitions, assumptions, and the energy-saving advantage of our proposed sensor control platform. Section 4 describes the two-phase clustering (TPC) scheme in detail. Section 5 shows how to realize reporting interval adaptive sensor control and data forwarding schedule. Sections 6 and 7 present simulation and implementation studies, respectively. Finally, Section 8 concludes the paper.

A preliminary version of this paper was presented at the 2004 Mobiquitous conference. This present paper is an improved version which has been extended to include supplementary algorithms and comprehensive simulation results

2. Related Works

Clustering techniques have been used to scale down the network and, with the help of its hierarchical structure, is considered as a promising technique to facilitate sensor control for extending network lifetime.

A clustering scheme, called low-energy adaptive clustering hierarchy (LEACH), proposed in [8] introduced a randomized cluster head rotation for single hop, wireless, sensor networks to uniformly distribute the energy consumption associated with the demands of being a cluster head. A small number of cluster heads is randomly chosen and other sensors choose the nearest cluster head. The cluster head sends the aggregated data to the base station, thus saving on energy consumption.

Table 1. Comparison of the proposed scheme to one-hop clustering schemes (e.g., LEACH, PEGASIS, TASC, and HEED) where CH and CM represent cluster head and member, respectively.

Scheme	Delay Adaptivity	CH Overhead	Forward Scheduling
Proposed Approach	Supported	Depending on delay requriements	Link-quality aware cross-layer scheduling
OCS	Not supported	Depending on the size of CM	N/A

More recently, a probabilistic, clustering algorithm, called HEED [14], has been proposed that periodically selects cluster heads based on a hybrid factor of the node residual energy level and nodal proximity. A topology adaptive, spatial clustering (TASC) scheme proposed in [15] forms clusters depending on node density variation. In [16], an adaptive cluster-based data collection scheme has been proposed, whereby sensors directly communicate with cluster heads and clusters are periodically generated in a random manner. The cluster formation is triggered by using a data sink.

The cluster head selection scheme [17] and its variant [18] allows each member to determine its own incentive, based upon the amount of energy thus far spent for data detection and transmission. The incentive value indicates how many rounds the member will exempt itself from consideration for becoming a cluster head. Depending on the suggested incentive by each member, the current cluster head reschedules the cluster head sequence that will be used in the next round. In addition, they are conditionally allowed to switch their cluster head depending on signal strength of their current cluster head.

[19] proposed a virtual cluster based energy efficient node wake-up scheduling for data collection and aggregation. The proposed scheme constructs a data gathering tree and forms virtual clusters, each with a parent node in the data gathering tree acting as the cluster head. Consecutive time slots are assigned to each cluster head depending on the number of child nodes, i.e., the more the child nodes the more are the time slots assigned. Each sensor changes its state only twice, i.e., *i*) sleep to transmit state and *ii*) sleep to reception state during the duration of consecutive time slots, and thus saving energy.

QoS-aware data reporting tree construction and QoS-aware node scheduling schemes have been proposed in [20]. A data gathering tree is constructed based on the end-to-end delay and the traffic load to find a routing path from each cluster head to sink while the proposed scheduling scheme selects a certain number of nodes in a cluster as a set of data reporters based on three given QoS parameters: *i*) data correlation, *ii*) energy level, and *iii*) data reporting frequency. Unselected sensors remain in a sleep state, thus saving energy.

All the schemes discussed above do not consider providing clusters with adaptivity to the reporting interval for further improvement on energy conservation depending on application-dependent or time-varying constraints. We call these clustering schemes one-hop clustering schemes (OCS). The main qualitative comparison of our proposed scheme with the OCS are summarized in **Table 1**.

3. Preliminaries

A wireless sensor network (WSN) consists of a large number of sensors, single or multiple base station(s), also called data sink(s), and wireless links representing direct communication between the sensors, or between sensors and a base station within the radio range. We shall define this WSN as an undirected graph $G = (V, E)$, where V is the set of nodes (sensors and base stations) and E is the set of edges (bidirectional wireless links). Each $s_i \in V$ has its maximum communication radio range with radius r_c and the area of s_i's radio range is denoted by A_{s_i}. The Euclidian distance and hop distance between a sender s_i and receiver s_j are denoted by $d(s_i, s_j)$ and $h(s_i, s_j)$, respectively. An edge, denoted by $e(s_i, s_j) \in E$, exists between s_i and s_j if $d(s_i, s_j) \leq r_c$. s_i working as a cluster head is denoted by c_i. A set $C \subseteq V$ is defined as a set of cluster heads. Each sensor $s_j \in \{V - C\}$ belongs to c_i. if and only if $d(c_i, s_j)$ is minimum among all the cluster heads in C. The set of cluster members of c_i will be denoted by M_{c_i} such that $\cup_{\forall c_i \epsilon C} M_{c_i} = V - C$. Now, let us introduce the following definitions.

Definition 3.1: The set of *local neighbors* of s_i is defined as $N_{s_i} = \{s_j | d(s_i, s_j) \leq r_c,\ i \neq j\}$.
Definition 3.2: Let $\hat{d}$ be the expected distance from s_i to $s_j \in N_{s_i}$. Since s_i can be anywhere within the radio range area A_{s_i}, using the polar coordinate system we have

$$\hat{d} = \int_0^{2\pi} \int_0^{r_c} x^2 f(x\cos\theta, x\sin\theta)\, \partial x \partial = \frac{2r_c}{3}, \tag{1}$$

where $f(x\cos\theta, x\sin\theta) = \frac{1}{\pi {r_c}^2}$ is the probability density function that s_j is located at a point within A_{s_i}.
Definition 3.3: A *direct link* of s_j is an edge $e(c_i, s_j) \in E$ between a cluster head c_i and s_j where $c_i \in C$ and $s_j \in M_{c_i}$. Each s_j has its own direct link used for conveying both sensed and control data.
Definition 3.4: A *data relay point* of s_j, denoted by rp_{s_j}, is a neighbor belonging to the same c_i. Let $T_{s_j} = \{s_k | d(s_j, s_k) < d(c_i, s_j),\ s_k \in \{(N_{s_j} - W_{s_j}) \cap M_{c_i}\}\}$ where $W_{s_j} \subseteq N_{s_j}$ is a set of neighbors which are already chosen as a data relay point for other cluster members (i.e., rp_{s_j} is not shared by any other neighbor once it is assigned to a sensor). Then rp_{s_j} is a neighbor s_k with $[d(s_j, s_k)]$, $\forall s_k \in T_{s_j}$. The relay points can conduct data aggregation using an aggregation function such as min, max, or average.
Definition 3.5: A *data relay link* is an edge $e(s_j, rp_{s_j}) \in E$ between s_j and rp_{s_j} where s_j, $rp_{s_j} \in M_{c_i}$. This link is only used for conveying sensed data. Every sensor s_j in M_{c_i} can have its own rp_{s_j} if it exists. A possible longest data relay pattern to deliver sensed data to c_i is: $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \cdots \rightarrow s_{|M_{c_i}|} \rightarrow c_i$.
Our proposed scheme is operated based on the following assumptions. However, we need to clarify that these are standard hypothesis adopted by many related research papers.

- Homogeneous sensors are uniformly distributed over a geographical area and they are capable of adjusting transmission power depending on the distance to the receiver as in [8].
- Each sensor is aware of its location in the network with the help of a GPS or other means as in [21][22].

- A sensed result collected by a cluster head has to be delivered to the base station over a multi-hop routing path if the base station is not within the radio range. We assume that a separate routing protocol such as a position-based routing protocol [23] runs in the routing layer.

3.1 Energy-Saving Advantage

It has been well recognized that the data transmission is the most energy-consuming among all operations of a sensor. Furthermore, the transmitted radio-frequency (RF) signals attenuate proportionally to the transmission distance and thus the signal power level at the receiver side, denoted by P_{rx}, is not the same as at the transmitter side ,denoted by P_{tx}. In general, this signal attenuation between the sender s_i and receiver s_j is modeled by the following equation [24]:

$$P_{rx} \approx \frac{P_{tx}}{d(s_i, s_j)^{\alpha}}, \qquad \text{for } (2 \leq \alpha \leq 4) \tag{2}$$

where α is the path attenuation exponent for a transmitted radio signal. Due to this distance-dependent power attenuation characteristics of RF signals, the shorter the transmission distance between two nodes the better is the energy-saving rate. However, it is not always true depending on link error probability. That is, when the link error probability is higher than a specific threshold, dividing a path into more hops could require more energy than a path with fewer hops (refer to the study in [25]).

Since cluster members can be located anywhere within the radio range of cluster head with the same probability, we calculate the expected distance from CH to CM using Eq. (1) assuming that cluster members are evenly distributed as shown in **Fig. 1**. We define notations used in this approximate energy-saving measurement in **Table 2**.

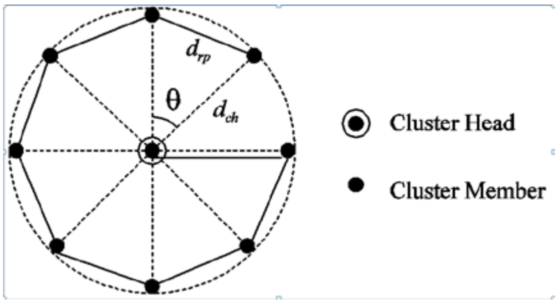


Fig. 1. Topology assumption to measure the energy-saving advantage. Solid lines represent data relay links and dotted lines represent direct links.

By introducing another type of link within clusters (clustering topology structure), we can increase the saving of energy in delivering sensed data to the cluster head. **Fig. 2** represents a linked transmission distance from all cluster members to cluster head. In our proposed scheme shown in **Fig. 2-(b)**, cluster members forward their sensed data to their data rely point except the last one, thus reducing the transmission distance and in the existing OCS schemes shown in **Fig. 2-(a)**, all cluster members forward data directly to cluster head.

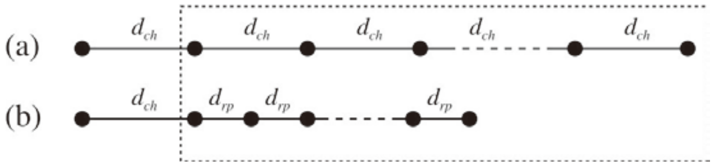


Fig. 2. Data forwarding pattern where d_{ch} and d_{rp} represent a distance to cluster head and relay point, respectively. **(a)** Existing OCS **(b)** Proposed scheme.

Considering a linked transmission distance only in a dotted line box in the **Fig. 2**, we can approximately measure the amount of energy savings in successfully forwarding sensed data to the cluster head.

Table 2. Notations

P_{ocs}	power consumption in the existing cluster schemes
P_{tpc}	Power consumption in our scheme
d_{ch}	distance to cluster head, i.e., $d(c_i, s_j)$, $s_j \in M_{c_i}$
d_{rp}	distance to relay point, i.e., $d(s_i, rp_{s_i})$, $s_i \in M_{c_i}$
p_e	link error probability in OCS
p_t	link error probability in TPC

There are $|M_{c_i}|$ cluster members in a cluster. Note that we only consider transmission distance and energy consumption caused by data transmission in the dotted box in **Fig. 2**. So, based on Eq. (2), total energy consumption in each cluster can be given as $P_{ocs} = \sum_{i=1}^{|M_{c_i}|-1} \frac{(d_{ch})^{\alpha} p_{rx}}{1-p_e}$and $P_{tpc} = \sum_{i=1}^{|M_{c_i}|-1} \frac{(d_{rp})^{\alpha} p_{rx}}{1-p_t}$ (assuming that the link error probability for all the hops is the same). Then, the energy-saving advantage of our scheme is given by

$$\frac{P_{ocs}}{P_{tpc}} = \frac{(d_{ch})^{\alpha}(1-p_e)^{-1}}{(d_{rp})^{\alpha}(1-p_t)^{-1}} = \gamma^{\alpha}\frac{(1-p_e)^{-1}}{(1-p_t)^{-1}} \tag{3}$$

where $\gamma = \frac{1}{2\sin\theta}$.

Finally, the energy-saving advantage (EA) is given as:

$$\text{EA} = \frac{P_{ocs}}{P_{tpc}} = \frac{\gamma^{\alpha}(1-p_e)^{-1}}{(1-p_t)^{-1}} \tag{4}$$

The above equation tells us that the energy-saving advantage (EA) is affected by the link error probability. Thus, consequently,

$$\text{EA} = \begin{cases} P_{ocs} = \gamma^{\alpha} P_{tpc} & \text{if } p_e = p_t, \\ P_{ocs} < P_{tpc} & \text{if } \frac{(1-p_e)}{(1-p_t)} > \gamma^{\alpha}, \\ P_{ocs} > P_{tpc} & \text{otherwise} \end{cases} \tag{5}$$

Eq. (5) shows our TPC-based sensed data transmission requires more energy consumption if $\frac{1-p_e}{1-p_t} > \gamma^{\alpha}$. The link error probability depends on how strong the received signal power is, when compared to the ambient noise level at the receiver side. This ambient noise is independent of the distance between the source and the destination. It is due mainly to environmental factors at the receiving moment, i.e., operating conditions [25]. Considering such a characteristic, the link error probability of each hop in our scheme is not much higher than that in the existing schemes, since the transmission power level is adjusted to be barely strong enough for the intended receiver in both cases. Definitely, there is a trade-off between energy-savings and message delivery latency.

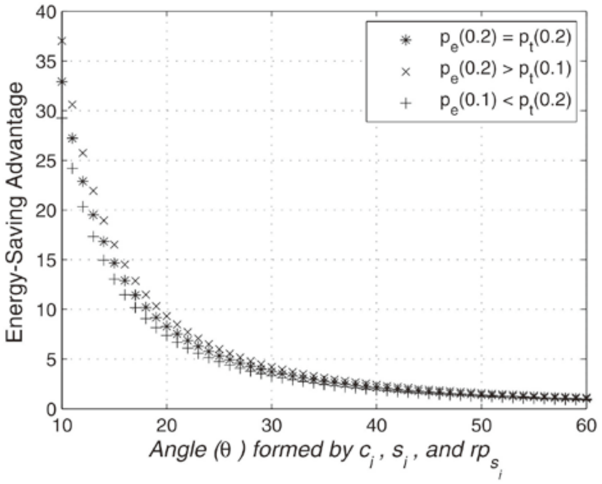


Fig. 3. Approximate energy-saving advantage of TPC-based data collection in each hop ($\alpha = 2$)

Based on Eq. (4), we numerically estimate the energy-saving advantage of our proposed scheme with varying number of cluster members which is represented by having varying angles as shown in **Fig. 1**. For example, if an angle θ is *20*, the number of cluster members becomes 18. **Fig. 3** shows that the denser the network, the larger the amount of energy-savings becomes.

4. TPC-based Sensor Control Platform

TPC forms clusters initially with closely located sensors in the phase I of cluster formation similar to other clustering schemes. Then, in phase II of the relay link construction, it attempts to restructure the intra-cluster node-connectivity in order to obtain further improvements in energy conservation. We describe how the two phases work in detail. For the details, refer to [26].

4.1 Cluster Formation (CF)

In phase I, the network is partitioned into clusters, each with a cluster head. We assume only one cluster head is allowed in a radio range. Since wireless communication cost is proportional to the transmission distance, the distance to the cluster head is chosen as a criterion for sensors to select their cluster heads, thereby minimizing energy consumption in forwarding sensed data directly to the cluster head (CH). In the case that there are more than two CHs with the same distance, a CH is randomly chosen to break the tie. Once a CH is chosen, sensors send a join-request to the CH to become a cluster member (CM). Upon overhearing a cluster head advertisement or a join-request message, s_i extracts the membership and location information of the sender from the message and stores them in its local neighbor set N_{s_i}. Now s_i can discern which neighbors belong to the same cluster, which ones serve as cluster heads, and their distance by querying N_{s_i}.

4.2 Construction of Relay Links (CR)

In phase II, clusters formed in phase I restructure the intra-cluster node connectivity by adding a set of additional edges (i.e., relay links) between cluster members. The purpose of phase II is two-fold — to minimize energy consumption in collecting sensed data from all the cluster members while meeting reporting interval constraints, and secondly to distribute the cluster head's workload. Every cluster member uses the data relay link instead of the direct link to send its sensed data to the cluster head when the given reporting interval is long enough. The sensed data conveyed along these relay links can be aggregated at each data relay point. The relay links are constructed at the cluster head's request to search for a data relay point. However, depending on the distribution and density of the sensors in the cluster, some cluster members may fail to find a data relay point such that a restructured intra-cluster node-connectivity results in a subgraph of a wheel graph centered at the cluster head, i.e., $0 \leq$ number of data relay points $< \left|M_{c_i}\right|$.

```
Procedure CR_CH begin    /* $\forall_{c_i} \in C$*/
 1: if $\{M_{c_i} - W_{c_i}\} = \emptyset$
 2: then stop and exit;
 3: else $n \leftarrow 1$; choose the farthest $s_i \in \{M_{c_i} - W_{c_i}\}$;
 4:     add $s_i$ to $W_{c_i}$ and send $m_{s_i}^{c_i}(n)$ to $s_i$;
 5:     monitor the construction procedure;
 6: end-if
 7: while (monitoring)
 8:     if $c_i$ overhears $m_{s_j}^{s_i}(n)$ then add $s_i$ to $W_{c_i}$; end-if
 9:     if $c_i$ receives $m_{c_i}^{s_i}(n)$
10:     then add $s_i$ to $W_{c_i}$;
11:         go to 1;
12: end-if
13: end-while
End-Procedure
```

Fig. 4. CH Procedure for Construction of Relay Links (CR)

The cluster head c_i launches the construction of relay links by choosing the farthest cluster member s_i in M_{c_i} as a starting point. c_i gives the starting point s_i a forwarding index, denoted by f_{ix}, which is initially set to 1. During the relay link construction, the f_{ix} increases by 1 whenever a data relay point is found. Whereas, the f_{ix} is reset to 1 if a cluster member fails to find its data relay point.

Two procedures for the relay link construction, CR_CH for c_i and CR_CM for all $s_i \in M_{c_i}$, are shown in **Fig. 4** and **5**, respectively. In both figures, W_{c_i} denotes a set of cluster members which have already completed the process of searching for a data relay point to set up a relay link; $m_{s_j}^{s_i}(n)$ represents a relay link setup message sent from s_i to s_j where $\mathrm{n} = 1 + f_{ix}$, in s_i.

Initially, a cluster member forwards the sensed data to its data relay point. The data relay point can be configured to aggregate its sensed data and then forwards the aggregated data to its data relay point, and so on. A data relay point conducts the aggregation only at its forwarding time and produces a data packet which in turn is forwarded to the next data relay point or the cluster head depending on the availability of the data relay point. Thus, data processing overhead remains the same regardless of the number of relays. The choice of using either the data relay link or the direct link to transmit the sensed data is controlled by the cluster head, depending on the reporting interval constraint given by the users/applications. The detailed description on link selection is given in Section 5.

```
Procedure CR_CM begin    /* $\forall_{s_i} \in M_{c_i}$ */
  monitor the construction procedure;
  while (monitoring)
    if $m_{s_i}^{c_i}(n)$ or $m_{s_i}^{s_j}(n)$ is received
    then $f_{ix} \leftarrow n$ as its forwarding index;
      find relay point $rp_{s_i} \in \{(M_{c_i} - W_{s_i}) \cap N_{s_i}\}$;
      if $rp_{s_i}$ exists
      then $n \leftarrow +1$;
        send $m_{rp_{s_i}}^{s_i}(n)$ to $rp_{s_i}$;
      else send $m_{c_i}^{s_i}(n)$ to $c_i$;
      end-if
    end-if
    if $m_{s_g}^{s_j}(n)$ or $m_{c_i}^{s_j}(n)$ is overheard    /* $s_g = rp_{s_j}$ */
    then add $s_j$ to $W_{s_i}$ if $s_j \in \{M_{c_i} - W_{s_i}\}$;
    end-if
  end-while
End-Procedure
```

Fig. 5. CM Procedure for Construction of Relay Links (CR)

5. Reporting Interval Adaptive Sensor Control

The user reporting interval requirement may vary depending upon the type of applications. Moreover, the state of such requirements can be either dynamic or static. For the dynamic case, the users control the sensor behavior (e.g., sensing or data reporting) by sending a control message depending on the environmental situation or the analysis of sensed data already delivered (such control information has been introduced as *interest* in [9]). For the static case, on the other hand, the sensors may have to decide on the data reporting interval based on a threshold value initially given. For a time critical situation, sensors should focus on meeting a reporting interval constraint even though energy consumption is relatively high. For example, sensors may sense a significant change in their sensing areas while monitoring a specific target or event. In such a case, the main concern is to deliver the sensed result to the users with the shortest delay. On the other hand, for a non time-critical situation such as periodic sensed data or network status reporting, energy-savings should have the highest priority for network longevity. Hence, sensors should be able to flexibly adapt to a time-varying or situation-dependent reporting interval and thus maximize energy conservation. In the following, we describe how a TPC-based data gathering platform helps us achieve the above requirements.

5.1 Decision on the Maximum Number of Relays

The TPC helps sensors maximize energy conservation by providing two types of data forwarding links — relay link and direct link — while they report sensed data to their local control center, i.e., cluster head. The steps for reporting interval adaptive sensor control are as follows:

1. CH calculates the possible maximum number of relaying sensors based on the reporting interval constraint given. MAC-dependent medium access delay, transmission delay, link quality, and sensor scheduling-dependent delay (e.g., sleep period) are major factors for the calculation.
2. CH broadcasts the possible maximum number of relaying sensors plus one as the seed, R_{max}, for CM to decide on the use of either "*direct link*" or "*relay link*".
3. CM decides which link to use by running *modulo* operation with the seed given by CH and its own forwarding index, i.e., f_{ix} mod R_{max}. If the result of this modulo operation is zero, use "*direct link*", otherwise use "*relay link*".

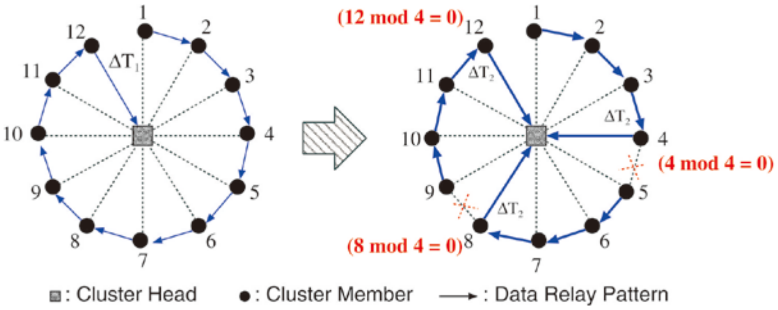


Fig. 6. Illustration of energy-saving reporting interval adaptive data gathering using TPC

Fig. 6 illustrates an example of how to adapt to varying sensed data reporting intervals using the data relay and direct links. In this example, the initial data reporting interval is $\Delta T_1 = 30$ seconds, i.e., the sensed data from all cluster members should be collected within 30 seconds. This initial interval is long enough for all cluster members to forward the sensed data via a relay link, thus achieving maximum energy conservation. While sensors perform a given task, the user dispatches a new reporting interval requirement, say $\Delta T_2 = 12$ seconds, to the cluster head, based on the analysis of the delivered sensed data. Upon receiving the reporting interval constraint the cluster head computes the maximum number of relays that can still meet the constraint. Let the result turn out to be 3 relays. The cluster head broadcasts a $R_{max} = 4$ control message. Then, the cluster members will forward their sensed data through either the direct link or data relay link, based on the result of f_{ix} mod 4. Sensors with $f_{ix} = 4$, 8, or 12 forward their sensed data (which is aggregated with sensed data received from the previous hop) to the cluster head using the direct link instead of data relay link since the result of the *modulo* operation is zero.

5.2 Considerations on Deciding R_{max}

Five major factors — medium access delay, transmission delay, probability of succesful transmission at the link-level (link quality), route discovery delay, and sensor scheduling delay — are mentioned that affect the calculation of the possible maximum number of relaying sensors. We denote them by M_d, T_d, L_s, R_d, and S_s, respectively. In particular, we define T_d as time delay until a packet is successfully delivered to the receiver side. We define a function that returns a relaying delay depending on the above five factors as:

$$f(M_d, T_d, L_s, R_d, S_s) \leq \Delta T_i$$

Since CH serves as a local control center, scheduling for medium access and sensor duty-cycle can be locally decided by harmonizing with other CHs. Among the above function parameters, M_d, R_d, and S_s are algorithm-dependent while T_d and L_s are environment-dependent, which means T_d and L_s are not as manageable as M_d, R_d, and S_s. In [27] where a transmission range is divided into three regions: connected, transient, and disconnected regions, it is shown that the link quality varies widely within its transmission range. We describe a way of handling this unstable link quality in calculating an expected relaying delay.

Depending on the link quality between two sensor nodes, the number of packet transmissions up to the first success, denoted by K_{data}, varies. We first model K_{data} as a Geometric random variable [28]. Given a link quality status, the probability that K_{data} becomes equal to k is measured by:

$$P[K_{data} = k] = (1 - L_s)^{k-1} L_s \qquad (6)$$

For the random variable K_{data}, the expected value of K_{data}, denoted by $E[K_{data}]$, is given by:

$$\begin{aligned} E[K_{data}] &= \sum_{k=0}^{\infty} kP[K_{data} = k] \\ &= \sum_{k=0}^{\infty} k(1 - L_s)^{k-1} L_s \\ &= \frac{1}{L_s} \end{aligned} \qquad (7)$$

So, if we define the function $f(M_d, T_d, L_s, R_d, S_s) = R_d + (M_d + T_d + S_s) \times E[K_{data}]$ and a packet should be delivered to the data sink, then

$$\Delta T_i \geq \sum_{j=1}^{T_{hop}} f_j(M_d, T_d, L_s, R_d, S_s) \qquad (8)$$

where T_{hop} is the total number of hops in delivering a packet to the data sink and is given by

$T_{hop} = R_{max} + hop(CH, dst)$, where $hop(CH, dst)$ is the number of hops from CH to *dst*.

So far, we have considered a packet transmission without using ACK confirmation. If sensors use ACK packet to confirm each successful packet transmission from the sender, the relaying delay function should be defined as $f(M_d, T_d, L_s, R_d, S_s) = R_d + (M_d + T_d + S_s) \times (E[K_{data}] * E[K_{ACK}] + E[K_{ACK}])$. **Fig. 7** illustrates the total number of packet transmissions to deliver a packet successfully between two sensor nodes in the case of using ACK.

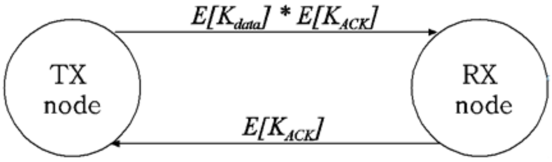


Fig. 7. Expected number of transmissions for successful data transmission in using ACK

In order to avoid a collision(s) while transmitting sensed data, TDMA protocol can be used. Moreover, due to the relatively low cost of implementation compared to TDMA protocol, CSMA protocol may be preferred as a MAC protocol and a CSMA-based virtual TDMA scheduling can be implemented. Sensors try to transmit what they acquired during a certain amount of given time and possibly enter into sleep mode. In this case, if the sensors fail to transmit for this given time slot, they should wait till their next turn or ignore this data transmission failure. To further reduce data transmission latency maintaining high data delivery ratio, a link quality-aware cross layer, data forwarding, scheduling scheme may be required. In the following, we show how to minimize a delay caused by M_d and S_d by using a cross-layer data forwarding scheduling scheme.

5.3 Cross-Layer Data Forwarding Schedule

All sensors within a cluster are now able to decide on whether they use either the direct link or relay link. Since the relay link is relatively shorter than the direct link, sensors can opportunistically save their energy resource.

In general, in a scheduled sensor network, sensors send data during a wake-up period and enter into sleep mode once the transmission is done. Depending on the type of sensors, some sensed data such as an image is large. So, appropriate amounts of time should be given to complete the transmission. Moreover, some applications require every data transmission to be guaranteed but others do not. The number of data transmissions considered to be a success depends mainly on the link quality between sender and receiver. Thus, we now introduce two possible data forwarding scheduling schemes which work with MAC scheduling — *uniform forwarding* and *probabilistic forwarding*.

5.3.1 Uniform Forwarding Scheduling

For uniform forwarding, all sensors in a cluster assume a receive-then-send behavior. Given ΔT_i, sensors with $f_{ix} = 1$ initiates forwarding the acquired data to their relay point at the beginning of ΔT_i. Once the relay points receive the sensed data, they relay the received data merged with what they sensed without any delay. After sending once, all sensors remains inactive until a new reporting interval starts. Every sensor has only one chance to forward data within ΔT_i. **Fig. 8** illustrates the forwarding scheduling of maximum n_{th} relayings where n_{th} relay point finally forwards the merged sensed data to the cluster head.

Note that each relay slot in this figure is a service-level time slot. In other words, each of these relay slots may contain multiple MAC-level time slots so that possible medium access collisions among sensors with $f_{ix} = 1$ is resolved in the MAC layer with an option of the local control of CH. The length of a relay slot, Δft_i, is given by $\Delta ft_i = f(D_s)$ where $f(D_s)$ is a function of the size of the data, D_s to be sent which returns the size of a service-level slot using the MAC-level time slot unit.

Sensors with no relay points directly forward sensed data to their cluster head. Note that a communication pair are both awake during Δft_i.

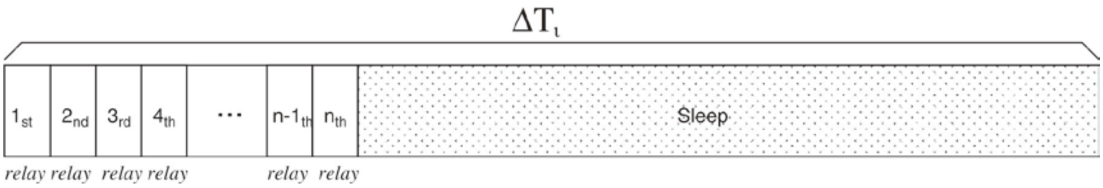


Fig. 8. Illustration of scheduling for uniform forwarding

5.3.2 Probabilistic Forwarding Scheduling

The wireless link quality is inversely proportionally affected by the distance [24]. As the distance increases, transmissions become prone to failures, so retransmission is required. Eq. (7) shows how the link quality effects the number of retransmissions. In the probabilistic forwarding, all sensors have their own relaying time slot to relay. The length of time is proportional to the distance between a sensor and its replay point (or cluster head). With this individual relaying time slot, the sensors have more time to complete relaying sensed data to the next relaying point. There are two cases in deciding the size of the time slot: *i*) expected successful data transmission with no ACK and *ii*) expected successful data transmission with ACK. For the case *i*), the number of data transmissions is $E[K_{data}]$ and for the case *ii*), it is $E[K_{data}] \times E[K_{ACK}] + E[K_{ACK}]$ as shown in **Fig. 7**.

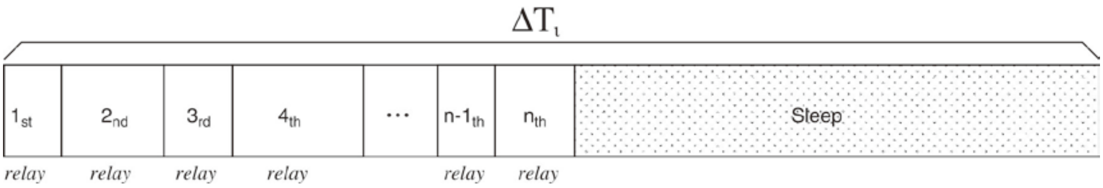


Fig. 9. Illustration of scheduling for probabilistic forwarding

Fig. 9 illustrates the forwarding scheduling of maximum n_{th} relays where each sensor has a different length for the relaying time depending on its distance to the relay point. The length of forwarding time slot, i.e., Δft_i, is calculated by:

$$\Delta ft_i = n(TX) \times f(D_s) \tag{9}$$

where $n(TX)$ is the number of data transmissions. For case *i*), $n(TX) = E[K_{data}]$ and $n(TX) = E[K_{data}] \times E[K_{ACK}] + E[K_{ACK}]$ for case *ii*).

Aforementioned, depending on sensor node distribution, node density may not be dense enough so that the resulting topology within a cluster becomes an incomplete wheel graph. In this case, there will be multiple sensors with $f_{ix} = 1$ and accordingly, there could be high collisions in accessing wireless medium to relay sensed data among the nodes. As illustrated in

Fig. 10, a slot considered in **Fig. 8** and **9** is a service-level time slot which includes multiple MAC-level slots. Thus, the collision can be removed by adding a MAC-level slot scheduling management onto cluster head. We do not describe the MAC level slot scheduling due to the space constraint of this paper.

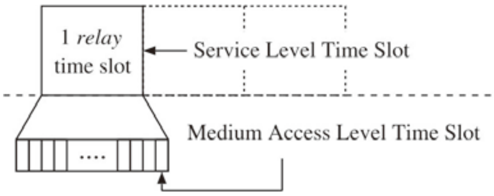


Fig. 10. Comparison of a service-level time slot with a MAC-level time slot

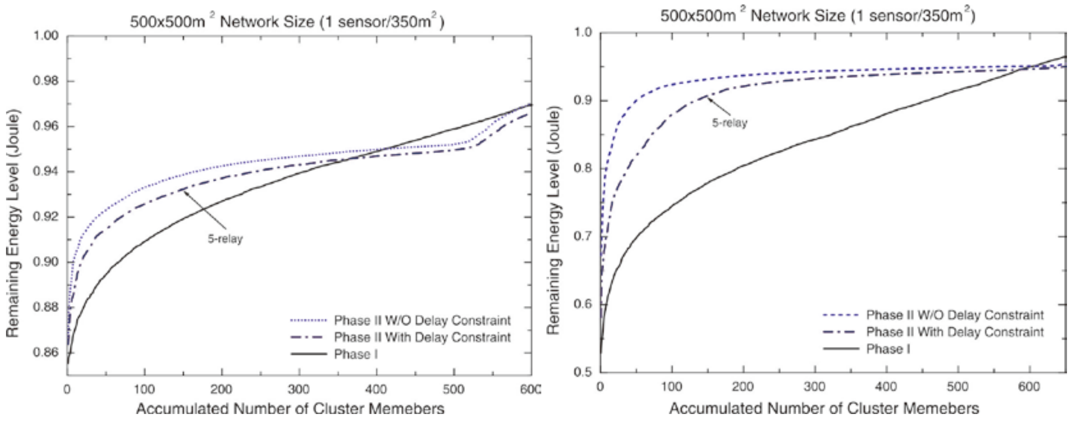
6. Simulation Study

We are primarily interested in studying the effects of phase II (*construction of relay links*) on conserving energy and on reducing the cluster head's workload. Although the network platforms that TPC and OCS are designed for are different, phase I (*cluster formation*) is a modified version of OCS with an enhancement in the context of multi-hop sensor networks. Therefore, we measure how much energy can be conserved by executing phase II and comparing it with the one achieved only by phase I. This provides a performance comparison of TPC with OCS.

6.1 Performance Metrics and Methodology

We evaluate the energy conservation capability of the TPC scheme by measuring the remaining energy level of each sensor after 1,000 reporting rounds to show how much energy can be saved. We conducted experiments using a JAVA thread-based implementation of TPC. In the experiments, homogeneous sensors are deployed in a $500m \times 500m$ network space based on a uniform distribution with 1 sensor/350m^2. Two communication radio ranges are used: $r_c = 50m$ and $100m$. After the completion of phases I and II, sensors start sensing an event and transmit sensed data to the cluster head with a data reporting interval of 3 seconds (i.e., continuous data reporting model). Sensors are set to generate the sensed data 1000 times. In order to measure the energy consumption for collecting sensed data from the cluster members, we used the same energy model as in [8] using radio electronics energy 50 nJ/bit, radio amplifier energy 100 pJ/bit, and 512 bit-size data packet (presented more details while we present the effect on energy-savings).

For the reporting interval adaptive data gathering experiment, we used a 5-*relay* control message. We performed five runs at each network density to collect the experimental results discussed below.



(a) **(b)**

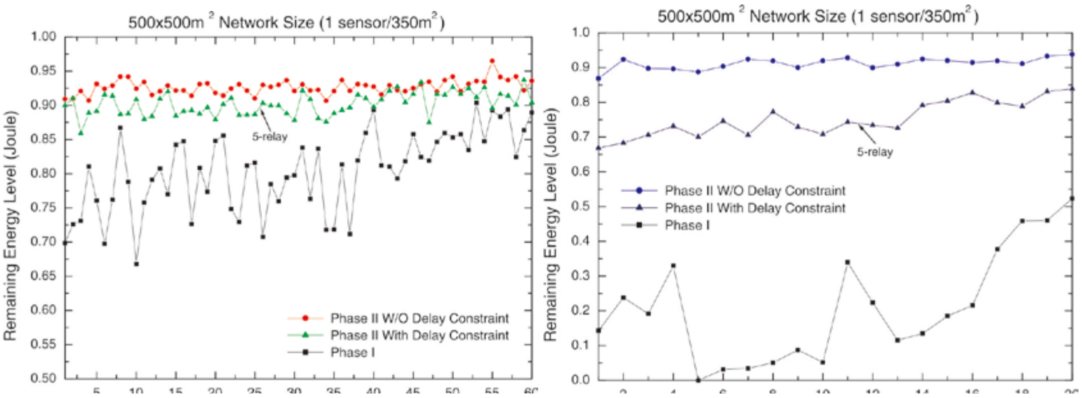
Fig. 11. Sorted remaining energy level of cluster members **(a)** $r_c = 50m$ and **(b)** $r_c = 100m$

6.2 Effects on Energy Savings and Cluster Head's Workload

In this section, we show the distribution of each sensor's remaining energy level after 1000 rounds of sensed data reporting (transmission). In general, for the low-radio wireless communication in free space, the communication distance between the sender and receiver as well as the electronics energy of the sensors are considered as the main factors in measuring energy consumption.

The total energy consumption for the sensed data collection within a cluster after phase II, depends upon the availability of rp_{s_i} in each cluster member. We use the energy model [8] to measure the energy consumption in our experiments for phases I and II and accordingly the remaining energy level in each sensor is computed. In order to show the overall trend of energy consumption, we sorted the remaining energy levels collected from each cluster member.

We set a maximum number of sensed data relays to 5 (i.e., $R_{max} = 5$) for reporting interval constraint data gathering. The radius of radio range $r_c = 50$ is used in **Fig. 11-(a)** and **12-(a)**, and $r_c = 100$ in **Fig. 11-(b)** and **12-(b)**. **Fig. 11-(a)** and **11-(b)** compare results of the remaining energy level of each sensor in phase II with and without a reporting interval constraint (i.e., 5-relay) and in phase I (direct communication with a cluster head). As mentioned previously, the radio range is a factor effecting the transmission distance which is proportional to the energy consumption. **Fig. 11-(b)** shows a much higher energy-savings compared to the one in **Fig. 11-(a)**. In order to meet the reporting interval constraint (5-relay), some cluster members require more energy using the direct link even though they have a data relay point. Thus, **Fig. 11-(a)** and **11-(b)** show that there is a trade-off between the data reporting interval and energy-savings. Unlike the direct communication with the cluster head in phase I, most of the cluster members in phase II need to consume some extra energy to receive data from a neighboring cluster member which chose them as a relay point, so that sensors closely located around the cluster head consume slightly more energy than in phase I. This explains the crossing points in **Fig. 11-(a)** and **11-(b)**.



(a) **(b)**

Fig. 12. Remaining energy level of cluster heads **(a)** $r_c = 50m$ and **(b)** $r_c = 100m$

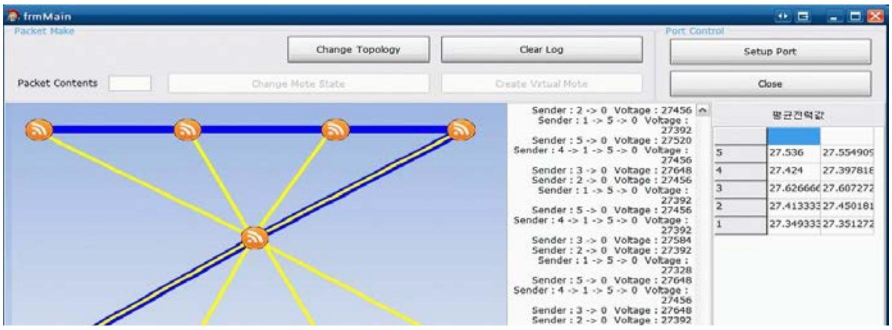
Fig. 12-(a) and **12-(b)** show a large difference in the remaining energy level of cluster heads between phases I and II. The results imply that there would be a much smaller number of cluster head rotations in the TPC scheme compared to protocols like OCS, in which cluster members communicate directly with the cluster head. This is because, in phase II very few sensors communicate directly with the cluster head since most of the cluster members forward their sensed data to their data relay point. Accordingly, the cluster head receives a smaller amount of sensed data. This implies that the overall workload of cluster head is reduced which depends on the number of cluster members directly communicating with the cluster head. As a result, we observe from **Fig. 12-(a)** and **12-(b)**, that the remaining energy levels of all the cluster heads in phase II are not much different from one another. The larger the radio range, the larger is the number of sensors in a cluster. This allows the cluster members to find a data relay point with high probability. Therefore, **Fig. 12-(b)** shows noticeable improvement in energy-savings in phase II compared to phase I. The plot for phase II with a 5-relay delay constraint reflects a trade-off between energy-savings and data reporting interval.

7. Implementation Study

We implemented our proposed reporting interval adaptive data gathering platform to more realistically evaluate its energy-saving efficiency, using a sensor development kit, called ZigbeX [13], from Hanback electronics [29]. **Table 3** shows the hardware specification of the used sensor motes. Due to the constraints of the space and the number of sensor motes we have at this moment, we were not able to fully implement the proposed sensor control platform forming a large sensor networks. **Fig. 13-(a)** illustrates the network topology we used in this implementation which includes 9 sensor motes.

Table 3. ZigbeX sensor mote specification

MCU	CPU: ATmega128L (7.3728Mhz), Flash memory 128KB, SRAM 4KB
RF Module	CC2420 (RF: 2.4GHz)
Flash Memory	512KB
Sensors	Temperature, Light, Photo Diode
OS	TinyOS 2.x

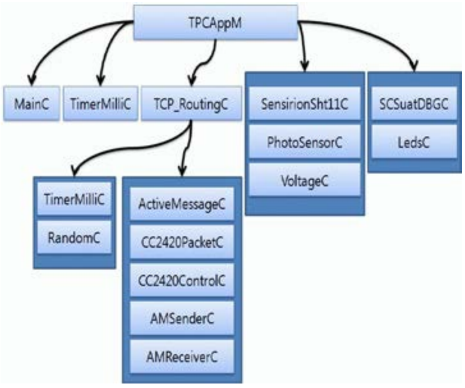


```
Sender : 2 -> 0  Voltage : 27456
Sender : 1 -> 5 -> 0  Voltage :
27392
Sender : 5 -> 0  Voltage : 27520
Sender : 4 -> 1 -> 5 -> 0  Voltage :
27456
Sender : 3 -> 0  Voltage : 27648
Sender : 2 -> 0  Voltage : 27456
Sender : 1 -> 5 -> 0  Voltage :
27392
Sender : 5 -> 0  Voltage : 27456
Sender : 4 -> 1 -> 5 -> 0  Voltage :
27392
Sender : 3 -> 0  Voltage : 27584
Sender : 2 -> 0  Voltage : 27392
Sender : 1 -> 5 -> 0  Voltage :
27328
Sender : 5 -> 0  Voltage : 27648
Sender : 4 -> 1 -> 5 -> 0  Voltage :
27456
Sender : 3 -> 0  Voltage : 27648
Sender : 2 -> 0  Voltage : 27392
```

평균전력값

5	27.536	27.554909
4	27.424	27.397818
3	27.626666	27.607272
2	27.413333	27.450181
1	27.349333	27.351272

(a)



(b)

Fig. 13. **(a)** Evaluation topology (8 cluster members + 1 cluster head) **(b)** Component architecture of TPC implementation

7.1 Architecture

The proposed platform was developed as an application running on top of TinyOS. **Fig. 13-(b)** shows the overall component architecture of the application. We describe the roles of main components in the figure.

- MainC: It maintains a data collection (reporting) interval, ΔT_i, and type of sensed data to be collected; it sends the collected data to a host PC which serves as the data sink.
- TimerMillic: Each sensor collects sensed data from sensors every timer event and passes it to TPC_RoutingC component.
- TPC_RoutingC: It constructs TPC and forwards sensed data passed by TimerMillic to either the relay point or the cluster head using either the direct link or relay link.

To form TPC, two beacon messages, *BeaconSink* and *BeaconNode*, were used. The BeaconSink from the cluster head was used for the initial clustering and controlling cluster members while the BeaconNode from the cluster members was used for constructing the relay links. By using the BeaconNode, the cluster member informs its neighbors of its cluster membership and current status. Hence, the cluster members can collect neighbor ID, link quality information (LQI), and the received signal strength indicator (RSSI) of the sender node which are required to search for the relay point.

7.2 Evaluation

Every 3000ms, sensor motes collect sensed data and transmit a 118 byte-sized packet. Depending on the collected RSSI value of the relay point, transmission power was adjusted based on 4 levels: 17.4mA, 15.2mA, 11.2mA, and 8.5mA. We statically set the maximum number of hops to the cluster head to 4 (i.e., the number of maximum relaying, R_{max}, is 4). The interval of the two beacon messages was set at 6000ms and for the maximum transmission power, 17.4mA power level was used. The minimum and maximum distance between two sensors are 3m and 8m, respectively.

To compare the amount of energy consumption of the two types of data gathering (i.e., OCS and TPC), we placed sensor motes in the same place and collected sensed data for 23 hours. Every hour, we measured the remaining battery level by using a voltage-meter. **Table 4** shows the remaining energy levels of TPC and OCS after our 23-hour data collection.

Table 4. Remaining energy level in both TPC-based and OCS-based sensor motes

	Mote 1	Mote 2	Mote 3	Mote 4	Mote 5	Mote 6	Mote 7	Mote 8
TPC	96.95%	96.96 %	96.92 %	96.99 %	97.12 %	96.95 %	97.01 %	96.94 %
OCS	89.23 %	88.81 %	89.20 %	89.69 %	89.17 %	89.38 %	89.36 %	89.13 %

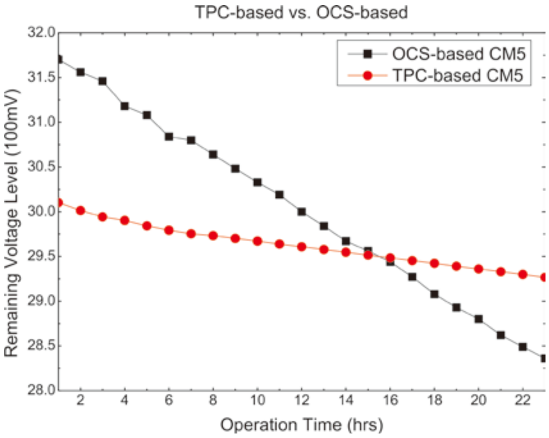


Fig. 14. Energy consumption comparison of TPC with OCS after 23-hour data collection

From the 23-hour data collection experiment, the average amount of energy consumption of TPC-based sensor motes is 0.09V while the one of OCS-based sensor motes is 0.341V. Consequently, it shows that the proposed TPC required only 26.75% energy consumption of the OCS. **Fig. 14** shows the comparison of TPC-based and OCS-based sensor nodes in terms of the remaining energy level. We selected a sensor mote since all the sensor motes have almost similar remaining energy levels. Note that initial energy levels for both cases, TPC and OCS, are different. The remaining energy level of the TPC-based sensor node goes down much more slowly than the one for the OCS-based sensor node. From this comparison, we can conclude that TPC definitely increases the network lifetime by reducing the energy consumption in each sensor yet without loss of the collected, sensed data.

8. Conclusion

In this paper, we introduced a reporting interval adaptive sensor control platform for energy-saving data gathering in wireless sensor networks. The proposed platform includes three components: *i*) a two phase clustering (TPC), *ii*) a reporting interval adaptive sensor control, and *iii*) a link, quality based, cross layer, data forwarding scheduling scheme. The goal is to provide sensors with high adaptability to the time- or situation-varying user reporting interval constraints in order to maximize energy conservation and hence the network lifetime. To achieve this goal, the proposed sensor control platform conducts the network clustering in two phases and provides a flexible sensor control mechanism and cross-layer data forwarding scheduling for reporting interval adaptive energy-saving data gathering. The network is partitioned into clusters, each with a *cluster head*, forming a direct link between the cluster head and each of the members. Each cluster member is required to search for a neighbor closer than its cluster head within the cluster to set up an energy-saving relay link. The direct link is used for managing cluster members and time-critical data collection.

The relay link is used for non-reporting interval constrained data collection forming a multi-hop data relaying capability. The use of the direct or relay link is flexibly controlled depending upon a given data reporting interval. Thus, sensors can opportunistically save their energy in reporting their sensed data. We also presented a reporting interval constraint. We considered those factors that should be evaluated in deciding the total number of relays and how the sensors should be scheduled depending on the link quality and the size of data to be sent for sensed data forwarding using the two types of links. Simulation and implementation studies demonstrate that the proposed sensor control platform can save a significant amount of energy for dense, sensor networks by making sensors adaptable to the user time-varying and application-specific reporting interval constraints.

References

[1] A. Ipakchi and F. Albuyeh, “Grid of the Future,” *IEEE Power and Energy Magazine*, vol. 7, no. 2, pp. 52–62, 2009. Article (CrossRef Link).

[2] E. A. Lee, “Cyber Physical Systems: Design Challenges,” in *Proc. of Int'l Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC)*, pp. 363-369, 2008. Article (CrossRef Link).

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, “Wireless Sensor Networks: A Survey,” *Computer Networks*, vol. 38, no. 4, pp.393–422, 2002. Article (CrossRef Link).

[4] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler and J. Anderson, “Wireless Sensor Networks for Habitat Monitoring,” in *Proc. of ACM Workshop on Wireless Sensor Networks and Applications (WSNA)*, pp. 88–97, 2002. Article (CrossRef Link).

[5] G. J. Pottie and W. J. Kaiser, “Wireless Integrated Network Sensors,” *Communications of the ACM*, vol. 43, no. 5, pp.51–58, 2000. Article (CrossRef Link).

[6] Y. Xu, J. Winter and W. Lee, “Dual Prediction-based Reporting for Object Tracking Sensor Networks,” in *Proc. of Int'l Conference on Mobile and Ubiquitous Systems: Networking and Services (MOBIQUITOUS)*, pp.154-163, 2004. Article (CrossRef Link).

[7] S. Ghiasi, A. Srivastava, X. Yang and M. Sarrafzadeh, “Optimal Energy Aware Clustering in Sensor Networks,” *Sensors*, vol. 2, no. 7, pp. 258–269, 2002. Article (CrossRef Link).

[8] W. B. Heinzelman, A. P. 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. Article (CrossRef Link).

[9] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," in *Proc. of ACM Mobile Computing and Networking (MOBICOM)*, pp. 56–67, 2000. Article (CrossRef Link).

[10] L. Krishnamachari, D. Estrin and S. Wicker, "The Impact of Data Aggregation in Wireless Sensor Networks," in *Proc. of IEEE Int'l Conference on Distributed Computing Systems Workshops (ICDCSW)*, pp. 575–578, 2002. Article (CrossRef Link).

[11] W. Choi, S. K. Das and K. Basu, "Angle-based Dynamic Path Construction for Route Load Balancing in Wireless Sensor Networks," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2474–2479, 2004. Article (CrossRef Link).

[12] X. Hong, M. Gerla, H. Wang and L. Clare, "Load Balanced, Energy-Aware Communications for Mars Sensor Networks," *IEEE Aerospace*, vol. 3, pp. 1109–1115, 2002. Article (CrossRef Link).

[13] "Ubiquitous Sensor Network System using ZigbeX," *3rd Edition. Hanback Electronics Research Center, Hanback. Electronics. Co.*, 2008. Article (CrossRef Link).

[14] 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.366–379, 2004. Article (CrossRef Link).

[15] R. Virrankoski and A. Savvides, "TACS: Topology Adaptive Spatial Clustering for Sensor Networks," in *Proc. of IEEE Int'l Conference on Mobile Ad hoc and Sensor Systems (MASS)*, pp. 1–10, 2005. Article (CrossRef Link).

[16] M. Lotfinezhad, B. Liang and E. S. Sousa, "Adaptive Cluster-Based Data Collection in Sensor Networks with Direct Sink Access," *IEEE Transactions on Mobile Computing*, vol. 7, no. 7, pp.884–897, 2007. Article (CrossRef Link).

[17] J. Baek, S. K. An, P. S. Fisher and E. J. Jones, "Dynamic Cluster Header Selection and with Self-incentive for Wireless Sensor Networks," in *Proc. of IEEE Sarnoff Symposium*, pp. 1–5, 2009. Article (CrossRef Link).

[18] J. Baek, S. K. An and P. S. Fisher, "Dynamic Cluster Header Selection and Conditional Re-clustering for Wireless Sensor Networks," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 4, pp.2249–2257, 2010. Article (CrossRef Link).

[19] Y. Wu, X. Li, Y. Liu and W. Lou, "Energy-Efficient Wake-Up Scheduling for Data Collection and Aggregation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 2, pp.275–287, 2010. Article (CrossRef Link).

[20] H. J. Choe, P. Ghosh and S. K. Das, "Cross-Layer Design for Adaptive Data Reporting in Wireless Sensor Networks," in *Proc. of IEEE Pervasive Computing (PERCOM)*, 2009. Article (CrossRef Link).

[21] N. Bulusu, J. Heidemann and D. Estrin, "GPS-less Low Cost Outdoor Localization for Very Small Devices," *IEEE Personal Communications*, vol. 7, no. 5, pp.28–34, 2000. Article (CrossRef Link).

[22] A. Nasipuri and K. Li, "A Directionality based Location Discovery Scheme for Wireless Sensor Networks," in *Proc. of ACM Workshop on Wireless Sensor Networks and Applications (WSNA)*, pp. 105–111, 2002. Article (CrossRef Link).

[23] I. Stojmenovic, "Position-Based Routing in Ad Hoc Networks," *IEEE Communications*, vol. 40, no. 7, pp.128–134, 2002. Article (CrossRef Link).

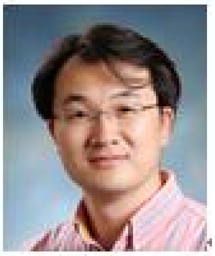
[24] F. Zhao and L. Guibas, "Wireless Sensor Networks: An Information Processing Approach," *Morgan Kaufmann*, 2004. Article (CrossRef Link).

[25] S. Banerjee and A. Misra, "Minimum Energy Paths for Reliable Communication in Multi-hop Wireless Networks," in *Proc. of ACM Mobile Ad-hoc Network Symposium (MOBIHOC)*, pp.145–156, Jun 2002. Article (CrossRef Link).

[26] W. Choi, P. Shah and S. Das, "A Framework for Energy-Saving Data Gathering Using Two-Phase Clustering in Wireless Sensor Networks," in *Proc. of Mobile and Ubiquitous Systems: Networking and Services (Mobiquitous)*, 2004. Article (CrossRef Link)

[27] M. Z. Zamalloa, K. Seada, B. Krishnamchari and A. Helmy, "Efficient Geographic Routing over Lossy Links in Wireless Sensor Networks," *ACM Transactions on Sensor Networks*, vol. 4, no. 12,

pp.1–32, 2008. Article (CrossRef Link).
[28] M. K. Molloy, "Fundamentals of Performance Modeling," *Macmillan*, 1989.
[29] Hanback. Electronics. Co. http://www.hanback.co.kr. Article (CrossRef Link).



Wook Choi currently works for System LSI Division, Samsung Electronics. He worked as an assistant professor in the Department of Computer and Engineering, Hankuk University of Foreign Studies in 2009 and 2010. He received his Ph.D. degree in Department of Computer Science and Engineering, University of Texas at Arlington in 2005. His research interests focus on USN, wireless mesh network, and routing protocols.

Yong Lee currently works as an assistant professor in the Department of Information and Communications Engineering, ChungJu National University. She received her Ph.D. degree in Department of Computer Science, Yonsei University in 2001. Her research interests include mobile communication, and wireless security.

Sangchul Kim currently works as a professor Department of Computer Science and Engineering, Hankuk University of Foreign Studies. He received his Ph.D. degree in Department of Computer Science, Michigan State University in 1994. His research interests include USN, multimedia systems, and serious game.