IEICE TRANSACTIONS

on Communications

VOL.E91-B
NO.3
MARCH 2008

A PUBLICATION OF THE COMMUNICATIONS SOCIETY

The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER

Proactive Data Filtering Algorithm for Aggregation in Wireless Sensor Networks*

Sungrae CHO[†a)], ***Member***

SUMMARY In this paper, proactive data filtering (PDF) algorithm is proposed for data aggregation (or data fusion) in wireless sensor networks. The objective of the algorithm is to further reduce the energy consumption when sensor nodes perform data aggregation. In many applications, the sensor field will be overwhelmed by unnecessary and redundant sensory information when the sink node disseminates a query throughout the sensor field. In order to reduce the energy consumption, our scheme employs intelligent decision logic in the sensor node which delays or deactivates the transmission of its response. A performance evaluation shows that data aggregation with the PDF significantly improves energy-efficiency.

key words: *data aggregation, data fusion, query, wireless sensor network*

1. Introduction

Wireless sensor networks have drawn strong attention recently from industries and research institutions as an enabling technology for invisible ubiquitous computing arena [16]. Spurred by the rapid convergence of key technologies such as digital circuitry, wireless communications, and micro electro mechanical systems (MEMS), a number of components in a sensor node can be integrated into a single chip which will reduce size, power consumption, and cost [1]. These small sensor nodes could be deployed in home, military, science, and industry applications such as transportation, health care, disaster recovery, warfare, security, industrial and building automation, and even space exploration. By connecting these small sensor nodes by radio links, the sensor nodes could perform tasks which traditional sensor nodes are hard to match.

Albeit the applications enabled by wireless sensor networks are very attractive, there are many technical challenges to overcome in order to build well-functioning robust system. The identified challenges include (1) *scalability*, (2) *adaptability*, (3) *addressing*, and (4) *energy-efficiency*. Since sensor networks consists of a large number of sensor nodes and thus large amount of data will be produced, large-scale data management techniques are needed.

Also, user constraints and environmental conditions, such as ambient noise, topology change, and event arrival rate, can be time-varying in wireless sensor networks. Thus, the system should be able to adapt to these time-varying conditions. Furthermore, sensor nodes may not have global identification because of the large amount of overhead and the large number of sensor nodes. Therefore, naming or addressing is a challenging issue in wireless sensor networks.

In addition to these challenges, the energy consumption of the underlying hardware and protocols is also of paramount importance. Wireless sensor nodes are expected to be operated by battery. Because of the requirement of unattended operation in remote or even potentially hostile locations, sensor networks are extremely energy-limited. Energy optimization in the sensor networks is much more complex since it involves not only reducing the energy consumption of single sensor node but also maximizing the lifetime of an entire network. The network lifetime can be maximized by incorporating energy awareness into every stage of wireless sensor network design and operation, thus empowering the system with the ability to make dynamic trade-offs between energy consumption, system performance, and operational fidelity [13].

Since various sensor nodes often detect common phenomena, there is likely to be some redundancy in the sensory data that the sources generate. In-network filtering and processing technique can therefore help to conserve the scarce energy resources. *Data aggregation* or *data fusion* has been identified as an essential paradigm for wireless routing in sensor networks [9]. The idea is to combine the data coming from different sources en-route — eliminating redundancy, minimizing the number of transmissions and thus saving energy.

In this paper, proactive data filtering (PDF) algorithm is proposed for data aggregation. The objective of the scheme is to further reduce the energy consumption when data aggregation** is involved. In order to reduce the energy consumption, our scheme employs an intelligent decision logic in the sensor node which defers or deactivates the transmission of its response.

The remainder of this paper is organized as follows. In Sect. 2, several existing data aggregation techniques and their problems are described. The proposed proactive data filtering (PDF) algorithm is given in Sect. 3. In Sect. 4, we compare the energy-efficiency and latency performance of data aggregation with or without our algorithm. Finally, contributions and future work are discussed in Sect. 5.

Manuscript received July 17, 2007.

[†]The author is with School of Computer Science and Engineering, Chung-Ang University, 221 Heukseok, Dongjak, Seoul 156-756, Republic of Korea.

*This work was supported by the MIC, Korea, under the HNRC-ITRC program supervised by the IITA.

a) E-mail: srcho@cau.ac.kr

DOI: 10.1093/ietcom/e91–b.3.742

**PDF algorithm can be used in any type of data aggregation protocol.

2. Data Aggregation Techniques in Wireless Sensor Network

Sensor nodes are scattered densely in a sensor field. A node called *sink* requests sensory information by sending a *query* throughout the sensor field. This query is received at sensor nodes (or *sources*). When the node finds data matching the query, the data (or response) is routed back to the sink by a multihop infrastructureless networked sensors. The information gathered in the sink node can be accessed by user via existing wide area networks such as Internet or satellite networks [1].

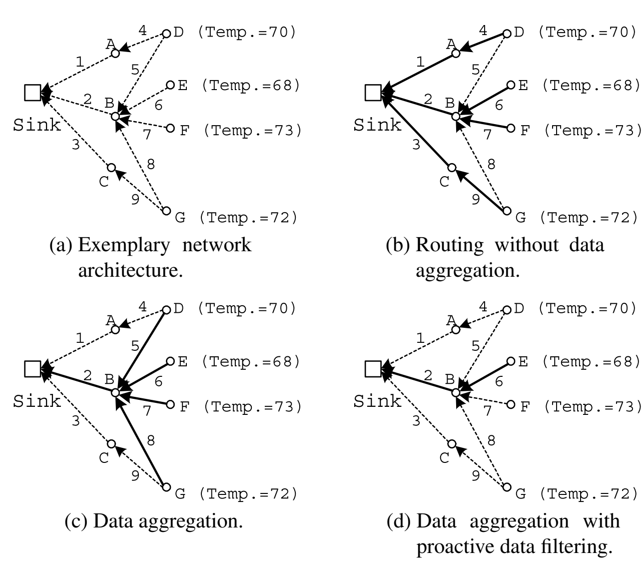
This query dissemination and sensory data gathering can be performed by the traditional *address-centric* approach where the shortest path is found based on physical end address as IP world has. The cost of an address in wireless sensor networks can be considered high if the address space is underutilized and the address space occupies greater portion of the total bits transmitted. Globally unique address would need to be very large compared to to typical size of data attached to them. Also, maintaining local address would be inefficient because more work is required to keep addresses locally unique as the network topology changes dynamically. In wireless sensor networks, a more favorable approach is the *data-centric* routing. In the data-centric approach, query dissemination is performed to assign the sensing tasks to the sensor nodes [1].

Data-centric routing requires an attribute-based naming [1], [12]. For the attribute based naming, the users are more interested in querying an attribute of the phenomenon, rather than querying an individual node. For instance, "*are there any node where the temperature is over 70 degree?*" is a more common query than "*what is the temperature measured by a certain node?*" The attribute-based naming is used to carry out queries by using the attributes of the phenomenon.

The key challenge in such sensory information gathering is conserving the sensor energy, so as to maximize their lifetime. *Data aggregation* (or *data fusion*) has emerged as a useful paradigm to reduce energy consumption in sensor networks. The key idea is to combine data from different sensors to eliminate redundant transmissions. By doing so, we can conserve entire network energy in collaborative fashion.

Intanagonwiwat et al. [7] discussed direct diffusion, a set of data-centric technique throughout the network. Their proposed operators are used to provide energy-efficient in-network data aggregation. Madden et al. [11] proposed TAG, an aggregation service as a part of TinyDB [15] which is a query processing system for a network of Berkeley motes. The service employs a SQL interface to the sensor data streams. It presents in-network processing of the aggregation queries on the data generated in the sensor network. Zhao et al. [19] introduced an architecture for sensor network monitoring. Their architecture benefits from an energy-efficient aggregate for network properties (digest functions). An average query is computed on the digest tree which their digest diffusion scheme constructs. Yuan et al. [18] introduced synchronization scheme for data aggregation when an event is detected at each sensor node (not triggered by a query). They proposed multi-level fusion synchronization (MFS) protocol which synchronizes the transmission time of each intermediate node so that the aggregation is performed effectively.

Most of the previous data aggregation techniques [5]–[10], [14], [17]–[19] aim at reducing the energy expended by the sensors during the process of data gathering. They form a hierarchical reverse tree topology from multiple source nodes to a sink where intermediate nodes filter or aggregate the redundant data from their child nodes. Thus, the aggregation is done in *spatial* rendezvous point. However, this spatial approach may not be energy-efficient since redundant data is still transmitted from the leaf nodes. An example is shown in Fig. 1. Suppose that the sink is interested in gathering a *minimum temperature* in the sensor field. Also, assume that each node can choose different route to report its measured data. For example, node D can select either link (or edge) #4 or #5 for transmission of sensory data as in Fig. 1(a). For simplicity, if we assume the same amount of energy is dissipated for transmission from one node to another, the energy budget per transmission is directly proportional to the number of links to be used for information gathering. Without data aggregation, the information gathering might look like as in Fig. 1(b). In this case, seven links are used in total to determine minimum temperature in the sensor field. With data aggregation, node B might be able to aggregate sensory information from nodes D, E, F, and G. As in Fig. 1(c), total number of links to be used is five. If we are able to employ a certain technique that only node with minimum temperature value can transmit its data, we require only two links to be used as in Fig. 1(d). Proactive data filtering (PDF) algorithm is motivated by this observa-



(a) Exemplary network architecture.
(b) Routing without data aggregation.
(c) Data aggregation.
(d) Data aggregation with proactive data filtering.

Fig. 1 Effect of different routing schemes (links to be used are highlighted by solid line).

tion.

Coverage of deployed sensors will overlap to ensure robust sensing task, so one event will likely trigger multiple sensors in the same phenomenon. In this case, it is likely to receive multiple identical copies of a sensory data. Also, some queries inherit redundant responses as follows:

- **Max:** The sink is interested in gathering maximum value from the sensor field. In this case, other values less than the maximum are redundant.
- **Min:** The sink is interested in gathering minimum value from the sensor field. In this case, other values greater than the minimum are redundant.
- **Existence:** Some application needs to identify the existence of a target object. For example, in directed diffusion [7], an initial query dissemination is used to determine if there indeed are any sensor nodes that detect the interested object.

We refer the query with the above types as a *singular* query which expects only one response from source nodes. Redundant and unnecessary responses will generate unnecessary transmission at the underlying layers. For example, unnecessary response will cause high duty cycle at the medium access control (MAC) layer which in turn generates high contention from multiple nodes. Consequently, sensor nodes suffer from unnecessary energy consumption.

The proposed PDF algorithm proactively suppress the redundant and unnecessary responses from sensor nodes so as to reduce the energy consumption. In the proactive data filtering (PDF) framework, a source node that receives a response from another source node will suppress its own local response if its local response is redundant. Responses are sent on an air interface to be received at other source nodes. If every source node defers its response by a random timer, a large number of redundant responses are likely to be suppressed. Similar idea called *snooping* was proposed in [11], but they did not consider the timing relationships among response transmissions from different sensor readings.

3. Proactive Data Filtering

Consider a network of tree rooted from a sink node to leaf nodes which is formed by some data aggregation scheme [5]–[10], [14], [17]–[19]. While the discussion of algorithms that help to generate and maintain this tree are beyond the scope of this work, our proposed filtering algorithm can function in any topology of the tree. We assume that the aggregation tree is formed at the network initialization phase, and is dynamically re-organized as sensors sleep, wake up, or fail.

Now, the depth of node_i is the number of edges from the sink to i. Suppose that the depth of the tree D is known *a priori* through a simple probing technique[†] in the network initialization phase. Also, we assume the propagation time from a node to another is negligible.

The proactive data filtering (PDF) algorithm is performed as follows:

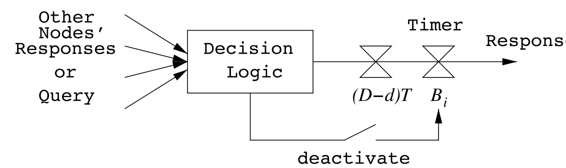


Fig. 2 A sensor node decision logic using proactive data filtering.

- The sink disseminates a query to its child nodes with (1) query type, (2) depth of the tree D, and (3) timer parameters[††] μ and T. The sink then waits for DT for responses from its child nodes.
- Each node simply forwards the query and waits for $(D-d)T$ where d is the depth of the node. By permitting this waiting time, each node is able to aggregate all the responses from its child nodes, and nodes in the same depth can be synchronized.
- When responses are received at source node_i from its child nodes during $(D-d)T$, it looks up the query type. If the query type is not singular, then the node immediately sends its response back to its parent after $(D-d)T$. If the query type is singular, it activates a timer after $(D-d)T$ with timer value B_i as shown in Fig. 2. Timer B_i is derived from received timer parameters μ and T. When timer expires, the source node transmits its response. If a response from other source nodes is received prior to timer expiration, node_i compares the received response with its own response. If node_i finds that its response is redundant, it deactivates its timer.

An important performance metric for the PDF algorithm is the expected number of responses since it is directly related to an energy budget. The expected number of responses, $E[\mathcal{R}]$ is given by

$$E[\mathcal{R}] = \sum_{j=1}^{R} E[\mathcal{R}_j] \tag{1}$$

where R is the number of nodes which match the query; and $\mathcal{R}_j$ is the number of responses at node_j and $E[\mathcal{R}_j] \leq 1$ since the node may or may not send its response.

$E[\mathcal{R}_j]$ is given by

$$\begin{aligned} E[\mathcal{R}_j] &= E\left\{I[\text{node}_j \text{ sends its response}]\right\} \\ &= P\left\{\text{node}_j \text{ sends its response}\right\} \\ &= P\left\{\bigcap_{\substack{i\in\Omega_j \\ i\neq j}} \begin{pmatrix} \text{node}_i \text{ does not suppress} \\ \text{the response of node}_j \end{pmatrix}\right\} \end{aligned} \tag{2}$$

where $I[\cdot]$ denotes the indicator function defined as $I[x] = 1$ if x is true; $I[x] = 0$ otherwise, and Ω_j is the set of nodes

[†]This can be easily done by sending a simple query from a sink where intermediate nodes simply increment the hop counter of the query. Then, the maximum hop counter will be the depth of the tree.

[††]These parameters are described in later part of this section.

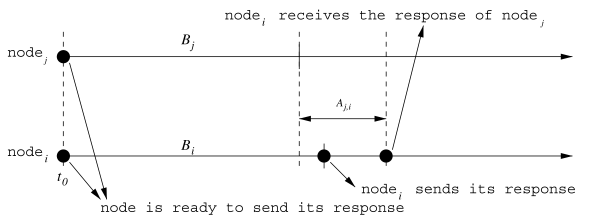


Fig. 3 Condition that $node_j$ does not suppress the response of $node_i$.

which are within communication distance from $node_j$.

Due to independence, (2) becomes

$$E[\mathcal{R}_j] = \prod_{\substack{i \in \Omega_j \\ i \neq j}} P\begin{pmatrix} \text{node}_i \text{ does not suppress} \\ \text{the response of node}_j \end{pmatrix} \tag{3}$$

Consider $node_i$ and $node_j$. Figure 3 illustrates the condition that $node_j$ does not suppress the response of $node_i$. Let B_i and B_j denote the timer for $node_i$ and $node_j$, respectively, where $0 \leq B_i \leq T$ and $0 \leq B_j \leq T$.

Suppose that nodes i and j are ready for sending their response at $t = t_0$ as shown in Fig. 3[†]. Once it is ready for sending its response, $node_j$ schedules its timer B_j while $node_i$ sets its timer as B_i. Suppose that $node_j$'s timer expires and sends its response at $t = t_0 + B_j$. Assuming $A_{j,i}$ denotes transmission delay from $node_j$ to $node_i$, $node_i$ will receive $node_j$'s feedback at $t = t_0 + B_j + A_{j,i}$. $node_i$ will suppress its response if $node_j$'s response is received before its timer expiration at $t_0 + B_i$. If $node_i$ sends out its response before receiving $node_j$'s response, then $node_j$ cannot suppress the $node_i$'s response. Therefore, the condition that $node_j$ does not suppress the response of $node_i$ is

$$t_0 + B_i < t_0 + B_j + A_{j,i} \tag{4}$$

or

$$B_i < B_j + A_{j,i} \tag{5}$$

From this observation, $P(\text{node}_i$ does not suppress the response of $\text{node}_j)$ is given by

$$\begin{aligned} &P\left(\text{node}_i \text{ does not suppress the response of node}_j\right) \\ &\quad = P(B_i < B_j + A_{j,i}) \end{aligned} \tag{6}$$

The PDF algorithm exploits random timer B_i. In order to avoid high contention during earlier portion of timer value, the weight of the timer density function moves toward later portion of B_i and results in a dense timer setting at high values of B_i. Hence, a relatively large number of sources willing to send their response will be suppressed. Hence, we use an exponentially distributed timer of which the probability density function is given by

$$f_{B_i}(b_i) = \begin{cases} \frac{\mu e^{\mu b_i}}{e^{\mu T}-1}, & \text{if } 0 \leq b_i \leq T \\ 0, & otherwise \end{cases} \tag{7}$$

Assuming the same delay $A_{i,j} = A_{j,i} = a$, (6) is simplified into

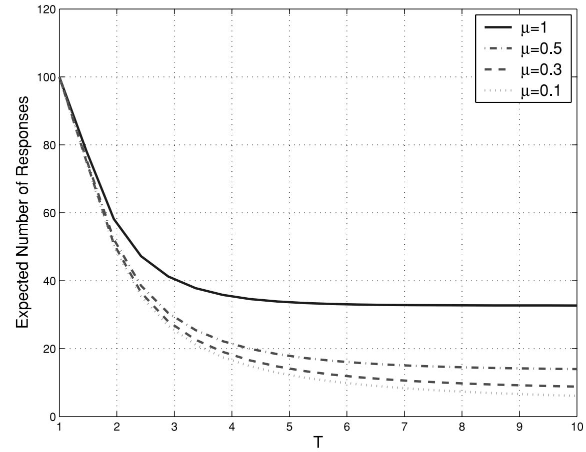


Fig. 4 The expected number of responses vs. T.

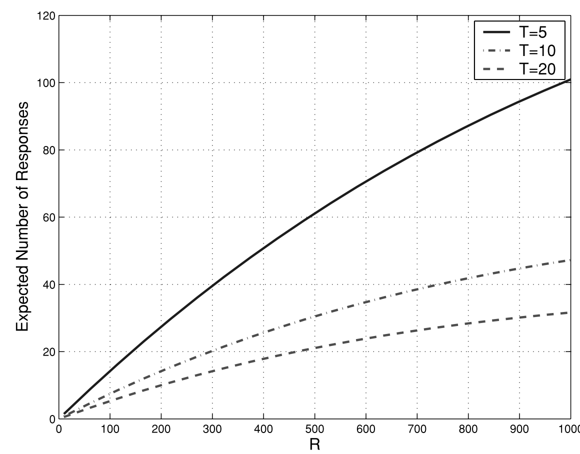


Fig. 5 The expected number of responses vs. R.

$$P\begin{pmatrix} \text{node}_i \text{ does} \\ \text{not suppress the} \\ \text{response of node}_j \end{pmatrix} = P(B_i - B_j < a) \tag{8}$$

or

$$\begin{aligned} &P\left(\text{node}_i \text{ does not suppress the response of node}_j\right) \\ &\quad = \int_0^T P(B_i < a + b_j) f_{B_j}(b_j) db_j \\ &\quad = \frac{2 + 2e^{2\mu T} - 2e^{\mu T} - e^{\mu(2T-a)} - e^{\mu a}}{2\left(e^{\mu T} - 1\right)^2} \end{aligned} \tag{9}$$

Figure 4 depicts the expected number of responses $E[\mathcal{R}]$ versus T with different parameters μ where $R = 100$. We observe from the figure that $E[\mathcal{R}]$ decreases with increasing T. With higher value of T, the slope is rather flat. We observe that at higher range of T (such as $T > 5$) and at $\mu \leq 0.5$, we achieve almost the small number of responses (less than 20).

In Fig. 5, we show the expected number of responses

[†]Since each node waits for $(D - d)T$ for its response, nodes in the same depth are synchronized.

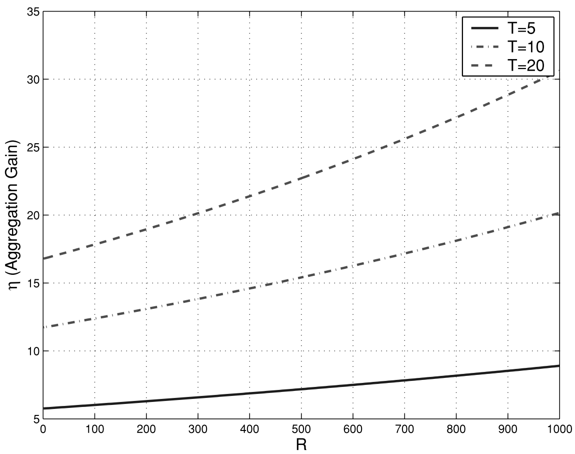


Fig. 6 Aggregation gain η vs. R.

versus the number of nodes R with different parameters $T = 5$, 10, and 20 time units. Here, we use $\mu = 0.1$ and $a = 1$ time unit. We observe that very small number of responses can be obtained with different parameter T under larger number of source nodes R (at most 101 responses at $T = 5$ and $R = 1000$).

Let $E[\mathcal{R}_0]$ and $E[\mathcal{R}_{PDF}]$ denote the expected number of responses without and with proactive data filtering (PDF) algorithm, respectively. Then, we define aggregation gain η as

$$\eta = \frac{E[\mathcal{R}_0] - E[\mathcal{R}_{PDF}]}{E[\mathcal{R}_{PDF}]} \tag{10}$$

which represents how small amount of energy can be dissipated using PDF algorithm. As can be seen in Fig. 6, the larger gain is achieved with larger number of nodes R. We also observe that higher value of T provides the larger aggregation gain.

Remark 1: We have shown that with PDF parameters μ and T the number of responses can be significantly reduced, i.e., energy-efficiency can be significantly improved. Especially, the more T we have, the more improvement in energy-efficiency we achieve. However, if we have a large T, the total latency in query processing increases. Therefore, the recommended choice of T would be $\arg\max_{t<D_{SINK}/D}\{t\}$ where D_{SINK} is the allowable maximum query latency at the sink and D is the depth of the tree. In the following section, we evaluate the performance of PDF algorithm with respect to energy-efficiency and latency in more realistic situation.

4. Performance Evaluation

To evaluate the energy-efficiency performance of PDF algorithm, we developed a simulator based on event-driven simulation using Java. The simulator generates a random topology as follows. We assume that the sensors have a fixed radio range and are placed in a square area randomly. Figure 7 shows a typical network routing tree. This tree

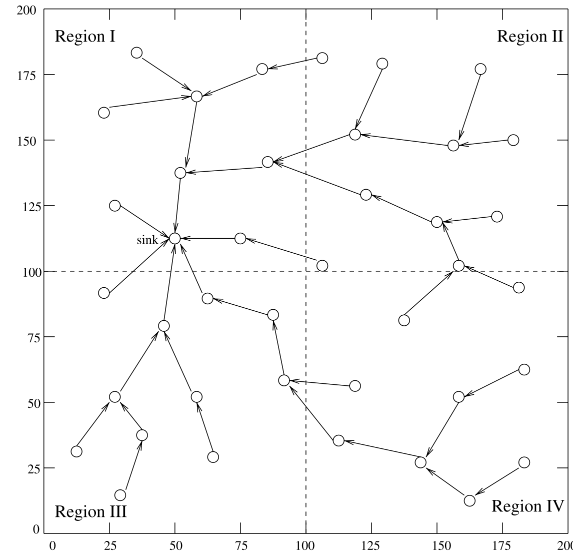


Fig. 7 An exemplary network routing tree for 40 nodes placed in a 200 × 200 area.

is formed based on the proximity metric of each node using breadth first search tree [3]. The root of the tree (sink) is randomly selected in the simulator. When we vary the number of sensors, we vary the size of the area over which they are distributed so as to keep the density of sensors constant. For instance, we use a 1000 × 1000 area for 1000 sensors. For 4000 sensors, the dimensions are enlarged to 2000 × 2000.

Based on the tree formed, the sink disseminates a query (with all query parameters as described in Sect. 3) to its child nodes which forward this query to their children. This process is continued until the query is reached to the deepest nodes. The depth of the tree is computed based on the tree formed, and is used for response waiting time ($(D - d)T$) at each of the node. When the query is reached to the deepest nodes, the deepest nodes respond with their sensor reading which are aggregated at their parent node, and so on towards the sink.

The sensor reading values are generated uniformly or non-uniformly. When uniform sensor reading is used, we generate the integer numbers in range of [10, 90] where we assume the minimum and maximum sensor readings are 1 and 100, respectively. In other words, the sink expects the sensor readings are from 1 to 100, but actual readings are between 10 and 90. The choice of the range is rather arbitrary, but we observed that expanding the reading range does not affect the performance when we also increase the node density. In real networks, the values of sensors are not uniformly distributed, but rather correlated with their location. For this reason, we consider non-uniform scenario in the simulation. In case of non-uniform sensor reading, we divide the area into four regions called Region I, II, III, and IV as shown in Fig. 7 with the following sensor reading ranges:

- In Region I, uniform sensor readings from 10 to 30,
- In Region II, uniform sensor readings from 30 to 50,

- In Region III, uniform sensor readings from 50 to 70, and
- In Region IV, uniform sensor readings from 70 to 90.

We performed the MAX query process as described in Sect. 2. In the simulation, we consider three protocols: (1) *aggregation without PDF and snooping*† (Algorithm 1), (2) *aggregation without PDF but with snooping* (Algorithm 2), and (3) *aggregation with PDF* (PDF). Algorithm 1 is a scheme in which nodes independently transmit their responses without considering other nodes' sensor reading. Algorithm 2 [11] is a technique that exploits other sensors' reading. In other words, node that receives other nodes' reading suppresses its transmission if that node finds its local sensor reading is redundant. In comparison with PDF, Algorithm 2 does not employ any timer, and thus will have a significant chance to have a large number of redundant responses.

In our simulation, the performance is measured according to the following metrics:

- *The number of responses transmitted*††*:* the total number of responses transmitted in the given network. The number of responses transmitted will be taken into consideration as an energy-budget, and
- *Latency:* the time interval from the instant the sink disseminates a query to the instant the sink receives the first response from sensors.

All performance data we present in this section is averaged over 12 different topologies. Since our work focuses on aggregation timing, we have not implemented any model for underlying layers such as physical layer and network layer. Therefore, we did not consider any error or delay at underlying protocols due to wireless channel impairments and contention. However, PDF algorithm uses the snooping mechanism, so we implemented in our simulator a minimum MAC function which can overhear other nodes' transmissions.

4.1 Energy-Efficiency

In this section, we demonstrate the energy-efficiency performance of three algorithms for data aggregation: Algorithm 1, Algorithm 2, and our proposed PDF. Figure 8 shows the number of responses transmitted in the network versus the number of nodes under uniform distribution of the sensor readings. As can be seen, the number of responses in all three algorithms linearly increases with the number of nodes but with different slope. We observe that PDF algorithm significantly reduces the number of responses. With a little more loose delay bound ($D_{SINK} = 50$), PDF algorithm improves the energy-efficiency (the number of responses) with about 50 % against PDF with $D_{SINK} = 20$ at $R = 1000$. It is because the more loose D_{SINK}, the more T we have. Even PDF with $D_{SINK} = 50$ improves the energy-efficiency with factor of 0.05 and 0.09 against Algorithm 1 and Algorithm 2, respectively.

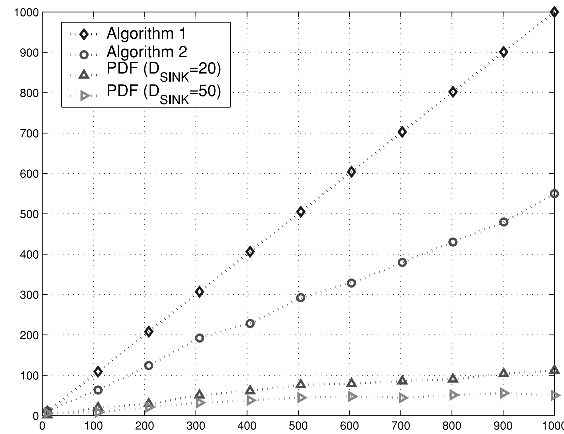


Fig. 8 Energy-efficiency performance vs. R (uniform distribution).

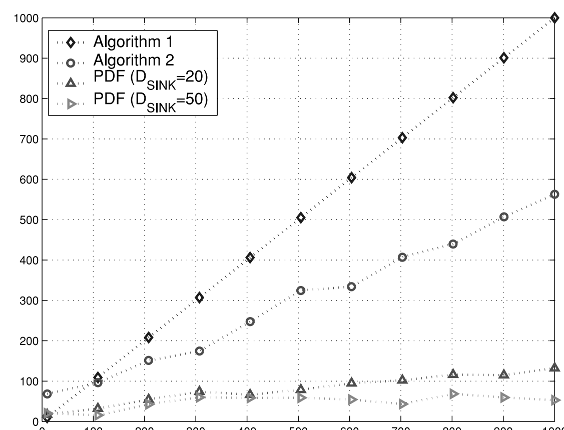


Fig. 9 Energy-efficiency performance vs. R (non-uniform distribution).

Figure 9 shows the energy-efficiency performance under non-uniform distribution of the sensor readings. The results are similar to the uniform case but with a little more fluctuation and with a slight increase of the number of responses. We conclude that the PDF algorithm outperforms Algorithms 1 and 2, independent of the distribution of the sensor readings.

To observe how power consumption is balanced among sensor nodes, we measure the number of transmissions per each node where 100 queries are generated, $D_{SINK} = 20$, and $R = 1000$ under uniform distribution of the sensor readings. Figure 10 shows the percent of the nodes whose response transmissions = N_T. As can be seen, the peak appears at $N_T = 11$. A certain node transmits 23 responses out of 100 queries as the maximum. Although the the number of transmissions per each node is not uniformly distributed, the

†Snooping is the mechanism that a node overhears other nodes' transmission. If the overhearing node finds its response redundant, then it suppresses its transmission.

††We measured the number of responses transmitted in the network instead of the number of responses received at the sink node. Counting the responses at the sink node only considers energy consumption at neighboring nodes of the sink.

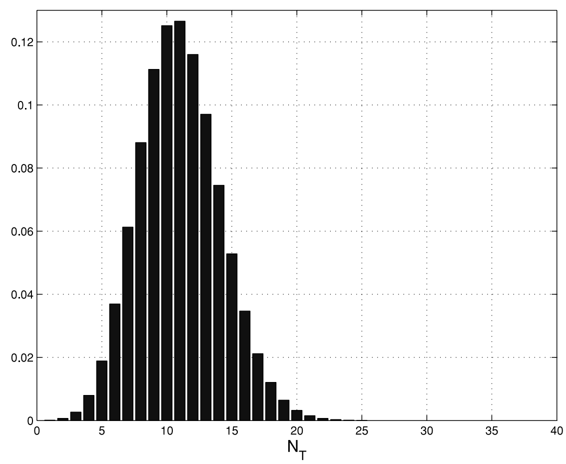


Fig. 10 Percent of the nodes of which response transmissions = N_T.

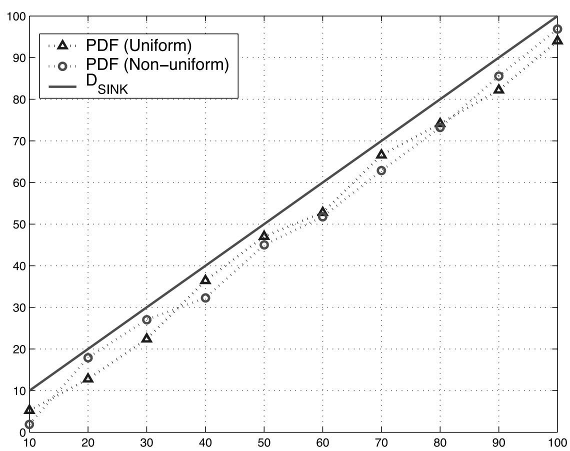


Fig. 11 Latency performance vs. R (uniform distribution).

PDF algorithm achieves significant reduction in power consumption, and the maximum number of response transmissions is observed as 23, which cannot be done without the PDF algorithm. (In Algorithm 1, the the maximum number of response transmissions is 100 out of 100 queries.)

4.2 Latency

Since the PDF imposes delay using random timer, we want to see the latency performance. In this section, we demonstrate the latency results of three algorithms. Throughout the simulation at $R = 100$, the latency of Algorithms 1 and 2 never exceeds 5 time units. But, Fig. 11 shows the latency performance of PDF versus D_{SINK} when $R = 100$ under uniform and non-uniform distributions of the sensor readings. As can be seen, the latency of PDF linearly increases with D_{SINK}. Although the latency performance shows that PDF algorithm inherently possesses significant delay compared with Algorithms 1 and 2, the latency never exceeds D_{SINK}. In many sensor monitoring systems, there should be the maximum allowable query latency as a system parameter. We argue that the delay within this latency bound can be considered acceptable. In our case, D_{SINK} is the allowable latency bound, and the delay in PDF algorithm always satisfies the bound.

Remark 2: The trade-off becomes clearer. We see that by trading off the delay subject to the maximum allowable query latency, we have more chance to suppress responses which results in energy-efficiency improvement.

5. Contributions and Future Work

In this paper, proactive data filtering (PDF) scheme is proposed. The objective of the scheme is to further reduce the energy consumption when the sink node collects sensory information from the nodes in the sensor field. In order to reduce the energy consumption, our scheme employ an intelligent decision logic in the sensor node which delays or deactivates the transmission of its response.

To our best knowledge, our work is unique and different from previous work. Perhaps, MFS [18] scheme would be the most comparable approach. However, their approach is based on the case that each sensor node detects an event (event-triggered report). Furthermore, they do not consider suppression of unnecessary duplicate responses.

Performance evaluation shows that data aggregation with PDF algorithm significantly improves energy-efficiency compared with other algorithms (up to factor of 1/20 against the scheme without data aggregation. This can be approximately interpreted as we can extend the network lifetime up to 20 times.). However, this improvement is done at the expense of the increased latency. Using simulation, we show that the latency always satisfied the maximum allowable latency bound. As as future work, we will investigate a mechanism to reduce the delay while providing similar level of energy-efficiency.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” IEEE Commun. Mag., vol.40, no.8, pp.102–114, Aug. 2002.

[2] P. Bonnet, J. Gehrke, and P. Seshadri, “Toward sensor database systems,” Proc. Mobile Data Management, pp.3–14, Jan. 2001.

[3] T.H. Cormen, C.E. Leiserson, and R.L. Rivest, Introduction to Algorithms, MIT Press, Cambridge, Massachusetts, 1990.

[4] S. Cho, “Bidirectional data aggregation scheme for wireless sensor networks,” Lecture Notes in Computer Science, vol.4159, pp.595–604, Sept. 2006.

[5] W.R. Heinzelman, J. Kulik, and H. Balakrishnan, “Adaptive protocols for information dissemination in wireless sensor networks,” Proc. ACM MOBICOM, pp.174–185, Seattle, WA, Aug. 1999.

[6] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” Proc. Hawaii International Conference on System Sciences, pp.1–10, Hawaii, Jan. 2000.

[7] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed diffusion: A scalable and robust communication paradigm for sensor networks,” Proc. ACM MOBICOM, pp.56–67, Boston, MA, Aug. 2000.

[8] B. Krishnamachari, D. Estrin, and S. Wicker, "Modeling data-centric routing in wireless sensor networks," USC Computer Engineering Technical Report, CENG 02-14, 2002.
[9] B. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," Proc. IEEE ICDCSW, pp.575–578, Rome, Italy, July 2002.
[10] S. Lindsey and C. Raghavendra, "PEGASIS: Power-efficient gathering in sensor information systems," Proc. Aerospace Conference, pp.1125–1130, Big Sky, MO, March 2002.
[11] S. Madden, S. Szewczyk, M.J. Franklin, and D. Culler, "Supporting aggregate queries over ad hoc sensor networks," Proc. Operating Systems Design and Implementation, Boston, MA, Dec. 2002.
[12] C.C. Shen, C. Srisathapornphat, and C. Jaikaeo, "Sensor information networking architecture and applications," IEEE Pers. Commun., vol.8, pp.52–59, Aug. 2001.
[13] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," Proc. ACM MOBICOM, pp.272–287, Rome, Italy, July 2001.
[14] N. Shrivastava, C. Buragohain, and D. Agrawal, "Medians and beyond: New aggregation techniques for sensor networks," Proc. ACM Sensys, pp.239–249, Baltimore, MD, Nov. 2004.
[15] http://tlelgraph.cs.berkeley.edu/tinydb
[16] M. Weiser, "The computer for the 21st century," Scientific American, vol.265, no.3, pp.94–104, Sept. 1991.
[17] Y. Yao and J. Gehrke, "The cougar approach to in-network query processing," Proc. ACM SIGMOD, pp.9–18, Madison, WI, June 2002.
[18] W. Yuan, S.V. Krishnamurthy, and S.K. Tripathi, "Synchronization of multiple levels of data fusion in wireless sensor networks," Proc. IEEE GLOBECOM, pp.221–225, San Francisco, Dec. 2003.
[19] J. Zhao, R. Govindan, and D. Estrin, "Computing aggregates for monitoring wireless sensor networks," Proc. IEEE SNPA, pp.139–148, Alaska, May 2003.

Sungrae Cho received the Ph.D. degree in Electrical and Computer Engineering from Georgia Institute of Technology, Atlanta, GA. in 2002 and the B.S. and M.S. degrees in Electronics Engineering from Korea University, Seoul, Korea in 1992 and 1994, respectively. He is currently an assistant professor in School of Computer Science and Engineering, Chung-Ang University (CAU), Seoul, South Korea. Prior to joining CAU, he was an assistant professor in Dept. of Computer Sciences, Georgia Southern University (GSU), Statesboro, GA from 2003 to 2006, and a senior member of technical staff at Samsung Advanced Institute of Technology (SAIT), Kiheung, South Korea in 2003. From 1994 to 1996, he was a member of research staff of Electronic and Telecommunication Research Institute (ETRI), Taejon, South Korea. His research interests include Wireless Sensor Networks, Wireless Mesh Networks, Cognitive Radio Networks, and Ubiquitous Computing.