Analysis of Optimized Aggregation Timing in Wireless Sensor Networks

Dong-Wook Lee and Jai-Hoon Kim
Graduate School of Information and Communication, Ajou University
San-5 Woncheon-Dong, YeongTong-Gu, Kyonggi-Do, 443-749, Korea
Phone: +82-31-219-2443, Fax: +82-31-219-1614
e-mail: {dwlee, jaikim}@ajou.ac.kr
*Corresponding author: Jai-Hoon Kim

Received January 22, 2009; revised February 27, 2009; accepted March 15, 2009; published April 25, 2009

Abstract

In a wireless sensor network (WSN) each sensor node deals with numerous sensing data elements. For the sake of energy efficiency and network lifetime, sensing data must be handled effectively. A technique used for this is data aggregation. Sending/receiving data involves numerous steps such as MAC layer control packet handshakes and route path setup, and these steps consume energy. Because these steps are involved in all data communication, the total cost increases are related to the counts of data sent/received. Therefore, many studies have proposed sending combined data, which is known as data aggregation. Very effective methods to aggregate sensing data have been suggested, but there is no means of deciding how long the sensor node should wait for aggregation. This is a very important issue, because the wait time affects the total communication cost and data reliability. There are two types of data aggregation; the data counting method and the time waiting method. However, each has weaknesses in terms of the delay. A hybrid method can be adopted to alleviate these problems. But, it cannot provide an optimal point of aggregation. In this paper, we suggest a stochastic-based data aggregation scheme, which provides the cost (in terms of communication and delay) optimal aggregation point. We present numerical analysis and results.

Keywords: Wireless sensor network, ubiquitous sensor network, data aggregation

This research is supported by the Ubiquitous Computing and Network (UCN) Project, Knowledge and Economy Frontier R&D Program of the Ministry of Knowledge Economy (MKE), the Korean government and a result of subproject UCN 09C1-C3-20S, the Korean government.

DOI: 10.3837/tiis.2009.02.006

1. Introduction

Recent advances in Micro-Electro Mechanical Systems (MEMS) have enabled the development of small sensor nodes composed of embedded devices such as communication boards and sensor boards. These sensor nodes have several types of sensing, computing and wireless communication capabilities. Sensor nodes can be deployed in an ad-hoc fashion, and operate via wireless communication. Because of these self-configuration abilities, the wireless sensor network (WSN) is attractive in many fields of research, e.g., military, environmental, etc. [1][2][3][4].

In WSN, several sensor nodes can detect the same sensing data at the position an event occurred. If they all transmit sensing data to the sink node, this results in numerous problems such as tremendous energy consumption, which shorten the network lifetime. Numerous steps are involved in sending the data (series of data packets),. In the MAC layer, a series of control packets must be sent/received to confirm/establish the next-hop receiver. For example, S-MAC [5] uses RTS/CTS packets to control the next-hop and sleep period. In the Network layer, the route path is set up first, prior to data transmission. Because these kinds of operations have a high energy consumption, each sensing data transmission causes discharging of the sensor node, resulting in network holes and reliability problems, and shortening of the network lifetime. Moreover, it generates heavy network traffic, which increases the end-to-end communication delay.

In order to solve this problem, many researchers have studied methods for transmitting combined sensing data which is known as data aggregation. By sending combined data, not only is the overhead caused by transmission diminished, but also the end-to-end delay is minimized. There are numerous types of data aggregation schemes [6][7][8][9][10][11][12]: centralized, tree-based, static-cluster-based, and dynamic-cluster-based. Each suggests a method to aggregate data, in terms of the aggregation path. However, although the proposed studies suggest very effective methods to aggregate sensing data, there is no means of deciding how long a sensor node should wait for aggregation. This is a very important issue, because the wait time affects the total communication cost and data reliability.

Some studies show how to optimize the end-to-end delay during data aggregation [13][14]. However, these methods are based on tree-based aggregation only, and the only metric used for optimization is the delay. So, the proposed optimized point is not the real optimized aggregation point.

In this paper, we suggest a stochastic-based data aggregation scheme, which provides the cost (in terms of communication and delay) optimal aggregation point. The optimal aggregation point is the point of minimum total cost. A node calculates the communication and delay costs according to the aggregation wait time. Based on the sum of these costs, a node determines the aggregation count of the minimum total cost. This results in reduced unnecessary wait time and enhanced utility of aggregation.

2. Related Work

There are numerous types of data aggregation schemes [15][16]: centralized, tree-based, static-cluster-based, and dynamic-cluster-based. Each scheme suggests how to aggregate data in terms of the aggregation path. A centralized scheme aggregates sensing data in the sink node. Every sensor node sends data without aggregation. After all data has arrived, the sink node aggregates the entire series of data. Despite the fact that this scheme is called aggregation, it is not in fact real aggregation. A tree-based scheme performs aggregation at the aggregation node [6], which is the merging point of two or more routing paths. If the network chooses a tree-based routing protocol, it constructs a data path tree first. And, ,there are some points in the tree structure at which different route paths are connected. At these points, aggregation can be performed. A static-cluster scheme is applied to a cluster network [7][8][9]. A cluster network consists of several clusters, which has a cluster header. In this scheme, every sensor node in the cluster sends its data to the cluster header, and it aggregates the received data. The operation of a dynamic-cluster scheme is similar to that of a static-cluster scheme [10][11], but its cluster is formed dynamically when an event occurs and it is sensed by nodes.

However, although each scheme suggests a very effective method to aggregate sensing data, we require a means of deciding how long sensor a node should wait for aggregation. Basically, there are two types of data aggregation that make these decisions: the data counting method and the time waiting method. In the data counting approach, each sensor node waits until receiving a predetermined number of packets. But, this method has problems such as the block state, which waits for the next packet forever. In the time waiting approach, each sensor node uses a timer, and if the timer is fired it sends the aggregated/non-aggregated data. But, this approach also has problems such as unnecessary waiting. A hybrid method can be adopted to alleviate the problems of both approaches, however, this is not the optimal solution. Some studies tried to determine the delay-optimal aggregation [12][13]. They involved performing data aggregation with a limited delay in tree-based aggregation. Each node calculates the wait time based on its level in the tree or the number of child nodes. These approaches are delay-optimized to some degree, but they are a kind of time waiting approach. Therefore, they also have similar problems such as unnecessary waiting.

Moreover, network traffic changes according to time and region. This means that to determine the optimal value, each node considers several factors such as the event sensing ratio and configuration values (i.e., limited delay, cost metrics).

In this paper, we suggest an optimal aggregation finding scheme, which considers not only the communication and delay cost, but also the ratio of generated sensing data.

3. Proposed Scheme

3.1 Basic Approach

We use two metrics: communication cost and delay cost. Based on these metrics, we can derive the cost function, which can provide the optimal aggregation point. If we assume that there is sufficient data flow, we can find the optimal aggregation point as follows.

First, we calculate the total cost when there is no aggregation (Eq. (1)).

$$Cost(t) = w_c \cdot C_c + w_d \cdot C_d(t) \tag{1}$$

$$(\mathbf{w_c}\text{: weight for communication, } \mathbf{w_d}\text{: weight for delay ,}$$
$$\mathbf{C_c}\text{: communication cost, } \mathbf{C_d}\text{: delay cost)}$$

Cost(t) means the total cost per packet when the node waits *t* seconds, *w* means the weighted value for each metric (i.e., communication cost and delay cost), and C_c is the communication cost. We assume that C_c is constant. In the real-world, C_c depends on numerous factors such as the bit-size of a packet, and the control packet handshake. However, we simply assume that C_c

is related to the sending data count. This is reasonable, because the data payload in WSN is very small compared to the packet header and control packet. As the data payload increases, C_c also increases. But, the packet header size and control packet handshake do not change. Therefore, the effect of increasing the data payload is negligible. C_d is the delay cost, and the wait-time related cost function, $C_d(t) = DelayCostFactor \times t$. We assume that the delay cost is propotional to the delay time and as the wait time increases the delay cost also increases. Based on Eq. (1), when both weighted values are equal ($w_c=w_d=1$), the total cost per packet, *Cost(t)*, is a wait-time related function. The total cost is linearly increased as the wait time increases, because there is no aggregation.

$$Cost(x) = w_c \cdot \frac{C_c}{(x+1)} + w_d \cdot C_d(x) \tag{2}$$

$$(x : aggregation\ count)$$

Cost(x) means the total cost per packet when the node aggregates x times (Thus, $(x+1)$-packets are reduced to one). We substitute the time t for the aggregation count x for ease of understanding. x is a function of t. In Eq. (2), the communication cost term is divided by $x+1$, because the aggregated packet is transmitted together. So, the cost per packet decreases by a factor of $x+1$. Otherwise, the delay cost term increases by a factor of x. This is because x additional packets must wait for aggregation. Thus, $C_d(x) = (DelayCostFactor \times t/2) \times x$. We assume that the weighted values are equal ($w_c=w_d=1$).

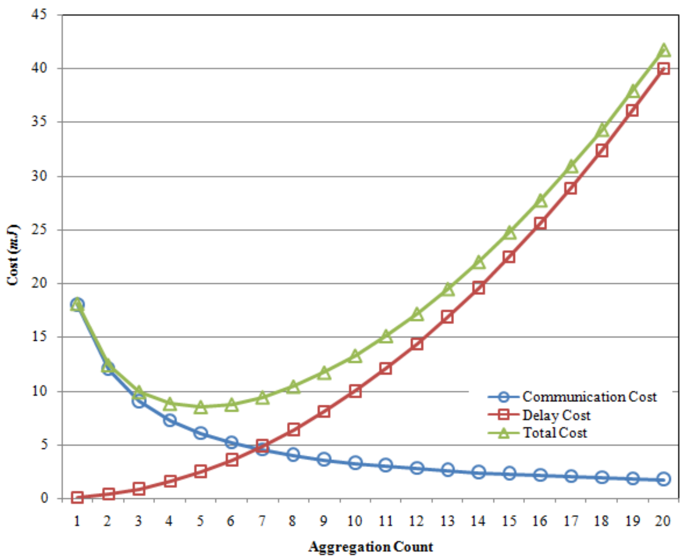

Fig. 1. Example of total cost of basic approach

Fig. 1 shows an example of the total cost of the basic approach. We set $C_c = 36\ mJ$ (two control packets , one data packet) [17], $DelayCostFactor = 0.1\ mJ/Sec$, $t=1/\lambda \times x$ (λ is the packet generation ratio) and $\lambda=0.5$. Based on **Fig. 1**, as the aggregation count increases, the communication cost decreases and the delay cost is increased. And, the total cost per packet is optimized ($8.5\ mJ$) when the aggregation count is 5. As a result, we can determine the optimal aggregation point in the given environments.

3.2 Stochastic Approach

In the basic approach, we present the core principle for optimizing the aggregation point. However, the real-world situation involves several factors. In **Fig. 1**, we assume that the packet generation ratio λ (which is the same as the event sensing ratio) is static. If $\lambda=0.5$, an event occurs every two seconds and a node senses the event. This assumption is unrealistic. So we use the following probability distribution function:

$$f(t) = \lambda \cdot e^{-\lambda \cdot t} \tag{3}$$

Based on Eq. (3), we can derive the success probability of aggregation during the t waiting time via integral calculus (0 to t). The success and failure probability is shown in Eq. (4).

$$\begin{aligned} P &= 1 - e^{-\lambda \cdot t} \ (\textit{success prob.}) \\ \overline{P} &= 1 - P = e^{-\lambda \cdot t} \ (\textit{failure prob.}) \end{aligned} \tag{4}$$

Also, we can derive the expected wait time $E(t)$,which is the time the node waits until the first successful aggregation on the condition of occurrence.

$$E(t) = \lambda^{-1} - \frac{t \cdot e^{-\lambda \cdot t}}{1 - e^{-\lambda \cdot t}} \tag{5}$$

So, we can calculate the total expected cost per packet of waiting for the first successful aggregation as follows:

$$\begin{gathered} ExpectedCost(t) = \overline{P} \times Cost\ of\ Failure\ Situation + P \times Cost\ of\ Success\ Situation \\ Cost\ of\ Failure\ Situation = C_c + C_d(t) \\ Cost\ of\ Success\ Situation = \frac{C_c}{2} + C_d(E(t)) \end{gathered} \tag{6}$$

Based on Eq. (6), the cost of the failure situation is merely one sending cost and the total t time waiting cost. The cost of the success situation is *50 %* of the sending cost (because of the cost per packet; two packets share the communication cost) and the $E(t)$ time waiting cost. So the $t - E(t)$ time waiting cost and *50 %* of the communication cost are saved.

Finally, we can derive the expected total cost per packet for the series of aggregations shown below. First, we calculate the total delay cost of the entire series of packets rather than that of each packet. The reason we only calculate the delay cost is that the communication cost is only incurred once, because of aggregation.

$$\begin{aligned} & WholeDelayCost_k(t) \\ & \quad = \overline{P_k} \times C_d(t) + P_k \times \left\{ k \times C_d(E_k(t)) + WholeDelayCost_{k+1}\left(t - E(t)\right) \right\} \end{aligned} \tag{7}$$

(k is initially set to one, and recursion is performed for aggregation count n)

The total delay cost means the total delay cost of an aggregation series. For each iteration, a delayed packet is added. Therefore, C_d must be multiplied by k. Eq. (8) shows the final expected cost calculation.

$$ExpectedCost(t)=\frac{C_c + WholeDelayCost_k(t)}{Expected\ Aggregation\ Count} \tag{8}$$

Whether aggregation occurs or not, only one transmission is required. This is also the case if there are several aggregations. But, the delay cost depends on the aggregation count. This is why we previously calculated WholeDelayCost in Eq. (7). And, since we want to derive the expected cost per packet, we divide the total cost by the expected aggregation count. The important point is that the expected aggregation count is not n, which is merely one case of aggregation. We want a stochastic-based approach that considers all cases. This is the reason for the expected aggregation count. The expected aggregation count can be used to derive the sum of the series of success probabilities.

By using a stochastic approach, we can determine the optimal point of aggregation based on the communication and delay costs. Also, each sensor node can calculate its own event sensing ratio (λ). Thus, every part of the sensor network can adapt the aggregation optimal point automatically.

4. Analysis

For analysis, we assume that there are three sensor nodes n_1, n_2, n_3 each with its own event sensing ratio, λ_1, λ_2, λ_3, respectively. We use the total event sensing ratio, $\lambda = \lambda_1 + \lambda_2 + \lambda_3$.

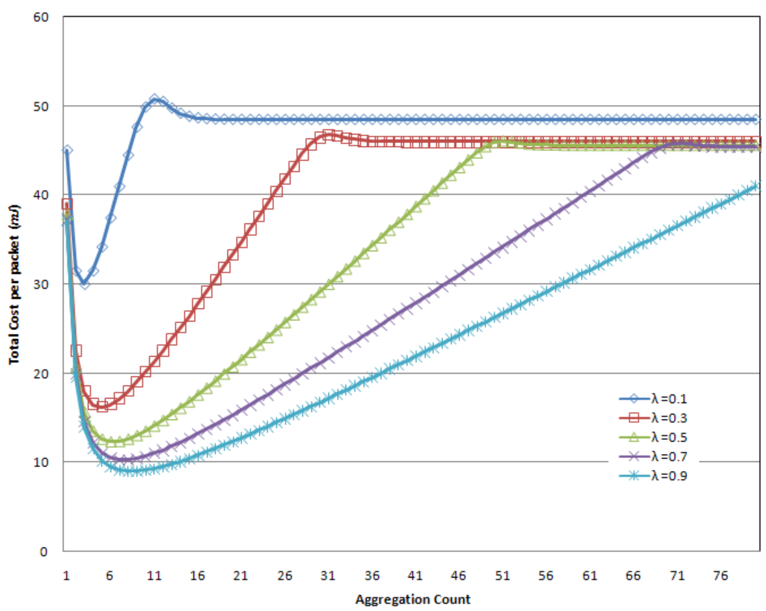

Fig. 2. Total cost per packet (C_c = 36 mJ, $DelayCostFactor$ = 0.9mJ/Sec)

Fig. 2 shows the result of analysis, where C_c = 36 mJ, $DelayCostFactor$ = 0.9 mJ/Sec. The event sensing ratio λ varies from *0.1* to *0.9*. Each graph in the figure has the lowest cost point . If λ=*0.1*, the lowest cost is *29.9994 mJ*, where the aggregation count is *3*. In each of the cases where λ=*0.3*, λ=*0.5*, λ=*0.7*, λ=*0.9*, the lowest cost is 16.2 *mJ*, *12.3 mJ, 10.2857 mJ, 9 mJ* respectively, and the aggregation count is *5, 6, 8, 9*, respectively. Thus, for each event sensing ratio, there is an optimal aggregation point (where the aggregation count is *3, 5, 6, 8, 9*, respectively) and each is different. Also, compared to the non-aggregation cost (*36 mJ*), every optimal point shows numerous gains. And, as λ increases, the amount of gain also increases. Compared to the case where λ = *0.1*, the case where λ = *0.9* saves more than *20 mJ*. In other words, if there are numerous events generated in the target field, the effect of data aggregation is maximized. But, beyond certain points, aggregation is less effective. i.e., for most cases in **Fig. 2**, beyond the aggregation count of *12* (where λ =*0.1*, 6), the cost exceeds *36 mJ*.

Table 1. Data sheet for Fig. 2.

Aggregation Count	Event sensing ratio (λ)				
	0.1	***0.3***	***0.5***	***0.7***	***0.9***
1	44.99959	39	37.8	37.28571	37
2	31.50093	22.5	20.7	19.92857	19.5
3	**29.9994**	18	15.6	14.57143	14
4	31.49415	16.5	13.5	12.21429	11.5
5	34.1781	**16.2**	12.6	11.05714	10.2
6	37.43119	16.5	**12.3**	10.5	9.5
7	40.94267	17.14286	12.34286	10.28571	9.142857
8	44.44976	18	12.6	**10.2857**	**9**
9	47.58821	19	13	10.42857	**9**
10	49.83186	20.1	13.5	10.67143	9.1
11	50.73137	21.27273	14.07273	10.98701	9.272727
12	44.99959	39	37.8	37.28571	37
…	…	…	…	…	…
80	48.43806	45.97176	45.56015	45.3946	40.94997

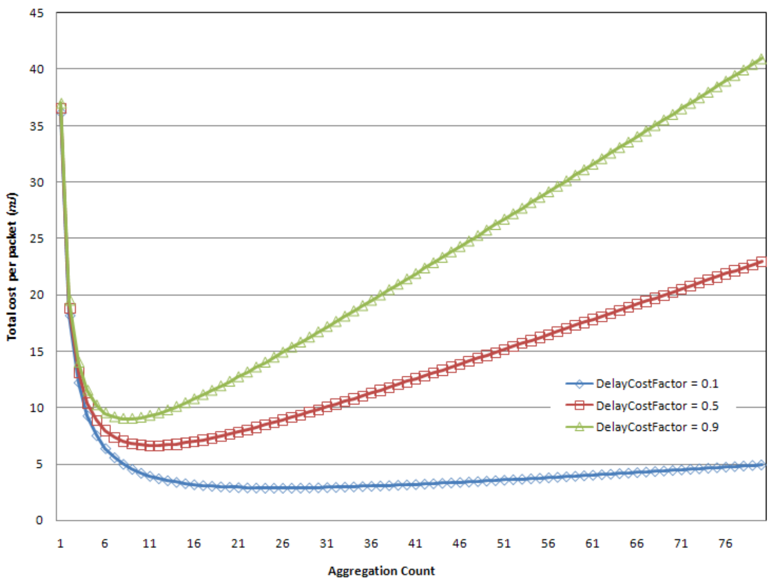

Fig. 3. Total cost per packet (C_c = *36 mJ*, λ = *0.9*)

Fig. 3 shows the cost graph in terms of the *DelayCostFactor* (*0.1, 0.5, 0.9*). As the *DelayCostFactor* increases, the optimal aggregation point decreases, and the cost increase factor is larger. Thus, if the *DelayCostFactor* increases, it is better to send data early rather than wait for aggregation.

Fig. 4 shows the cost graph in terms of the communication cost C_c (*18, 36, 72*). The optimal aggregation point differs slightly (*6, 8, 12*). As C_c increases, more aggregation is effective. This is because several packets share the communication costs. Moreover, in contrast to **Fig. 3**, as the aggregation count increases, the cost becomes increasingly similar. Thus, data aggregation is more sensitive to the delay cost than the communication cost. This is because the communication cost is constant, whereas the delay cost depends on time.

Based on the analysis, it is clear that the higher the number of events sensed, and the higher the communication cost, the more effectively aggregation is optimized. Also, if the delay cost is large, it is more effective to send data as soon as possible.

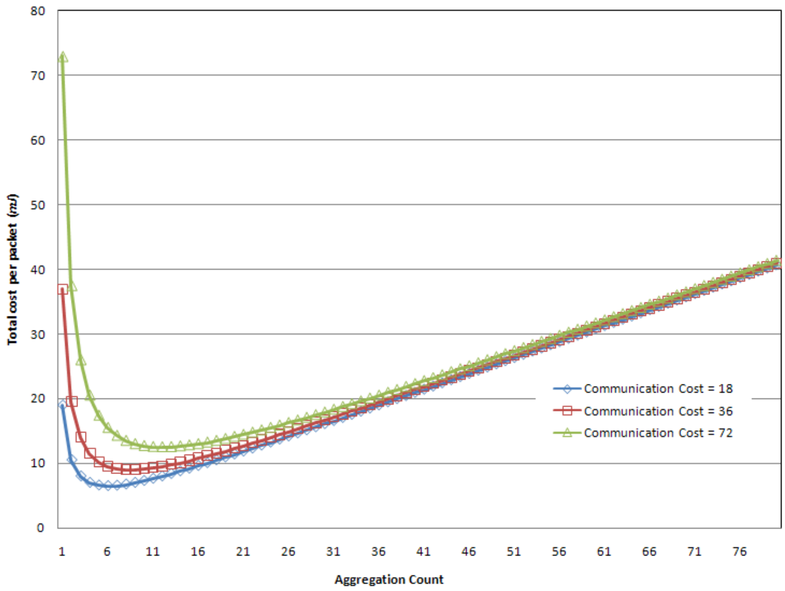

Fig. 4. Total cost per packet ($\lambda = 0.9$, *DelayCostFactor* = *0.9mJ/Sec*)

5. Conclusions

In this paper, we provided a method of finding the optimal aggregation point based on cost. For the sake of energy efficiency, network lifetime and reliability, data aggregation is inevitable in WSN. Data aggregation combines several data transmissions, which saves communication energy and reduces network traffic. Also, it prevents discharging of the sensor node, which eliminates network holes and increases data reliability. The important issue in data aggregation is deciding long the node should wait for aggregation. The static methods such as data couting and time waiting are unsuitable, because numerous metrics such as the event sensing ratio, delay contraint and traffic dependent communication cost depend on time. By using our proposed scheme, which considers these kinds of metrics, each node can find its optimal aggregation point, which is adaptive to envionmental changes. Therefore, each part of the sensor network can perform data aggregation specific to its particular enviroment, and this enables a more practical sensor network.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Communication Magazine*, pp.6-28, Aug. 2002.

[2] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," in *Proceeding of WSNA 2002*, Atlanta, Georgia, USA, pp.88-97, Sep. 2002.

[3] A. Ailamaki, C. Faloutos, P.S. Fischbeck, M.J. Small and J.V. Briensen, "An environmental sensor network to determine drinking water quality and security," in *SPECIAL ISSUE: Special section on sensor network technology and sensor data management, SIGMOD*, vol.32, no.4, pp.47-52, Dec. 2003.

[4] A. LaMarca, W. Brunnete, D. Koizumi, M. Lease, S.B. Sigurdsson, K. Sikorski, D. Fox and G. Borriello, "Making sensor network practical with robots," in *Proceedings of the First International Conference on Pervasive Computing, LNCS*, vol.2414, pp.152-166, 2002.

[5] W. Ye, J. Heidemann and D. Estrin, "An energy efficient MAC protocol for wireless sensor networks," in *Proc. IEEE INFOCOM* 2002, vol.3, pp.1567-1576, June 2002.

[6] B. Krishnamachari, D. Estrin and S.B. Wicker, "The impact of data aggregation in wireless sensor networks," in *ICDCSW '02: Proceedings of the 22nd International Conference on Distributed Computing Systems*, pp.575-578, 2002.

[7] M. Busse, T. Haenselmann and W. Effelsberg, "TECA: A Topology and Energy Control Algorithm for Wireless Sensor Networks," in *ACM Proceedings of Modeling analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, Oct. 2006.

[8] Y. Chen and S.H. Son, "A fault tolerant topology control in wireless sensor networks," in *ACS/IEEE International Conference on Computer Systems and Applications (AICCSA)*, Jan. 2005.

[9] R. Ghosh and S. Basagni, "Napping backbones: energy efficient topology control for wireless sensor networks," in *IEEE Radio and Wireless Symposium (RWS)*, Jan. 2006.

[10] W.P. Chen, J.C. Hou and L. Sha, "Dynamic clustering for acoustic target tracking in wireless sensor networks," in *IEEE Transactions on Mobile Computing*, vol.3, no.3, pp.258-271, July 2004.

[11] J.Y. Cheng, S.Ruan, R.Cheng and T.Hsu, "PADCP: Power-Aware Dynamic Clustering Protocol for Wireless Sensor Network," in *IFIP International Conference on Wireless and Optical Communications Networks (WOCN)*, Apr. 2006.

[12] Y. Yang, X. Wang, S. Zhu and G. Cao, "SDAP: A Secure Hop-by-Hop Data Aggregation Protocol for Sensor Networks," *ACM Transactions on Information and System Security*, vol.11, no.4, pp.1-43, July 2008.

[13] F. Hu and C. May, "Timing Control During Data Aggregation Operations of Distributed Sensor Networks: Trade-off Between Query Accuracy and Response Delay," *Information Technology Journal*, vol.5, no.5, pp.759-778, 2006.

[14] H. Li, H. Yu, B. Yang and A. Liu, "Timing control for delay-constrained data aggregation in wireless sensor networks," *International Journal of Communication Systems*, vol.20, pp.875-887, 2007.

[15] S. J. Park, "Performance Analysis of Data Aggregation Schemes for Wireless Sensor Networks Static-cluster aggregation and Dynamic-cluster aggregation Section," A dissertation for Doctoral in North Carolina State University, 2006.

[16] E. Fasolo, M. Rossi, J. Widmer and M.Zorzi, "In-network Aggregation Techniques for Wireless Sensor Networks: A Survey," *IEEE Wireless Communications*, vol.14, no.2, pp.70-87, Apr. 2007.

[17] Crossbow MICA2 mote Data Sheets, http://www.xbow.com

Dong-Wook Lee received BS and MS degrees in computer engineering from Ajou University, South Korea, in 2004 and 2006, respectively. He is currently a Ph.D. student of the Graduate School of Information and Communication at Ajou University. His research interests include wireless sensor networks and ubiquitous computing.

Jai-Hoon Kim received a BS degree in control and instrumentation engineering from Seoul National University, South Korea, in 1984, and an MS degree in computer science from Indiana University, U.S.A., in 1993, and a Ph.D. degree in Computer Science from Texas A&M University, USA, in 1997. He is currently a professor of the Graduate School of Information and Communication at Ajou University, South Korea. His research interests include distributed systems, mobile computing and ubiquitous computing.