社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
NS2006-9(2006-4)

[奨励講演]センサネットワークのための集約率を考慮した GIT 経路制御の評価

青西 孝文† 芳野 宏徳† 三上 真司†† 太田 能††† 川口 博††† 吉本 雅彦†††

† 神戸大学大学院 自然科学研究科 情報知能工学専攻
†† 金沢大学大学院 自然科学研究科 電子情報科学専攻
††† 神戸大学 工学部 情報知能工学科
E-mail: †{tkfm,yoshino,mik,ohta,kawapy,yosimoto}@cs28.cs.kobe-u.ac.jp

あらまし センサネットワークのための経路制御の研究では，完全集約が仮定されている場合が多くみられる．そのような仮定は，センサネットワークのアプリケーションを制約する．本稿では，GIT 経路制御において集約効率がエネルギ消費に及ぼす影響を検討する．われわれは，集約効率に応じた最適な集約点を選択することで，どの程度のエネルギ消費を削減できるのかに興味がある．本稿では，GIT を改良した AGIT（Aggregation efficiency-aware GIT）方式を提案する．また，exploratory message を抑圧するための方法として， hop exploratory 方式を検討する．シミュレーション結果より，AGIT は GIT に比べてデータ転送時のエネルギ消費を削減できることを示す．
キーワード センサネットワーク，集約率，GIT 経路制御方式

[Encouragement Talk]Evaluation of GIT Routing Considering Aggregation Ratio in Sensor Networks

Takafumi AONISHI†, Hironori YOSHINO†, Shinji MIKAMI††, Chikara OHTA†††, Hiroshi KAWAGUCHI†††, and Masahiko YOSHIMOTO†††

† Graduate School of Science and Technology, Kobe University, 1-1 Rokkodai, Nada, Kobe, 657-8501
†† Graduate School of Natural Science and Technology, Kanazawa University, Kakuma Kanazawa, 920-1192
††† Faculty of Engineering, Kobe University, 1-1 Rokkodai, Nada, Kobe, 657-8501
E-mail: †{tkfm,yoshino,mik,ohta,kawapy,yosimoto}@cs28.cs.kobe-u.ac.jp

Abstract In most research work for sensor network routings, perfect aggregation has been assumed. Such an assumption might limit the application of the wireless sensor networks. We address the impact of aggregation efficiency on energy consumption in the context of GIT routing. Our question is the extent to which energy consumption can decrease compared to the original GIT. In this paper, we propose an improved GIT: "aggregation efficiency-aware routing", or AGIT. We also consider a suppression scheme for exploratory messages: "hop exploratory." Our simulation results show that the AGIT saves the energy consumption of the data transmission compared to the original GIT.
Key words sensor networks, aggregation efficiency, greedy incremental tree routing

1. Introduction

Sensor networks are expected to operate under severe energy constraints because it is not practical to replace their batteries because of the large number of sensor nodes. A salient issue is reduction of the amount of transmitted data because wireless communications at sensor nodes consume more power than any other activity [8–12].

Data centric routing is a promising paradigm for sensor network routing [10]. With data centric routing, routing decisions are based on the contents of the payloads of packets rather than their destination addresses. A sensor node might aggregate receiving packets that are temporally buffered, generate a new packet, and then send it to the next hop. Such a means of operation is expected to reduce the amount of transmitted data, engendering remarkable power savings. An example of data centric routing is directed diffusion (DD) [7].

In most studies, perfect aggregation has been assumed (e.g. [3, 5, 8, 9]). In this case, the most efficient data paths from sources to a sink form a Steiner tree and/or minimal spanning tree. This fact encourages research of heuristic distributed algorithms such as Greedy Incremental Tree (GIT) [8] and the Nearest Neighbor Tree (NNT) [9]. However, perfect aggregation is not universal and possibly limits applications of sensor networks, as mentioned above. Unfortunately, we do not have sufficient insight into the influence of the diversity of the aggregation to sensor network routings.

In this work, we address the impact of aggregation efficiency on the energy consumption in the context of the GIT routing [8]. The original GIT is a heuristic algorithm to find a Steiner tree on a hop-count basis. Our questions are how the most efficient incremental aggregation point changes according to aggregation efficiency and how much energy consumption can decrease compared to the original GIT. we improve the GIT routing algorithm to find a more efficient aggregation point according to aggregation efficiency. In this paper, we call the improved GIT "aggregation efficiency-aware GIT (AGIT)."

This paper is organized as follows: In Section 2., we propose AGIT. Section 3. shows some simulation results. Finally, we present conclusions in Section 4..

2. Aggregation Efficiency-Aware GIT

2.1 Suppression of Exploratory Messages

In the DD, which is the basis of GIT routing, exploratory messages are distributed widely according to the nodes' gradients because interests do not contain any information about a sink. As a result, the gradients are set in many directions. (See Section 2.2.2 in [7].)

To some extent, GIT-like routing necessarily distributes exploratory messages in order to determine the aggregation point for the existing path tree. Results of our analysis showed, however, that the aggregation point becomes nearer to the sink than the foot of the perpendicular from the additional source to the existing path in the case of $\frac{1}{2} < r$. But, since the aggregation point becomes nearer to the sink, the path tree will become similar to that of "opportunistic routing", where data from different sources can be opportunistically aggregated at intermediate nodes along the established paths.

In AGIT, we consider the following a scheme to suppress the excessive exploratory messages: "hop exploratory." In the following, we assume that each node can know the hop-count from the sink through interest dissemination. Each node caches the hop-count from the sink for each interest as "*own_hop*." To do so, we also assume that each interest has a random identifier to be distinguished from the others.

2.1.1 Hop Exploratory

In addition to the field *previous_hop*, each exploratory message contains the additional field "*hop*" to store the hop count from the source that initiated the exploratory message. Whenever a source initiates the exploratory message with both *previous_hop* and *hop* set to *own_hop*.

When the node receives the exploratory message with *previous_hop*, it rebroadcasts the exploratory messages with *previous_hop* set to *own_hop* and with *hop* decremented by one if

$$own\_hop \leq previous\_hop \quad \text{and} \quad hop > 0. \tag{1}$$

2.2 Adjustment of the Incremental Cost Message Phase

The above suppression scheme involve some adjustments of the incremental cost message phase because the source nodes on the existing path tree might not receive the exploratory messages. Consequently, the incremental cost message is issued in such a case.

We take the following approach to overcome this problem. The intermediate nodes aside from the sources on the existing path tree can initiate the incremental cost message. In order to suppress the multiple incremental cost message, the more distant intermediate node from the sink issues the incremental cost message earlier. To do so, each intermediate node sets up an incremental cost message timer as

$$t_i = (max\_hop - own\_hop) \times \delta, \tag{2}$$

where max_hop and δ respectively denote the predefined network diameter and the timer granularity. The intermediate node issues the exploratory message if its timer expires before receiving another exploratory message; otherwise it suspends the issue.

2.3 Finding of Optimal Aggregation Point

In the following, we assume that linear aggregation is employed, whereby a packet has size $L_{\text{packet}} = L_{\text{header}} + L_{\text{payload}}$ where L_{header} is the header length and L_{payload} is the payload length in bytes. Furthermore, we assume that all sources send the data packet at the same rate. The procedure shown here can be extended easily to function in different cases.

In AGIT, the incremental cost message contains an additional field to store the hop-count H from an interim aggregation point. Whenever the source and/or the intermediate nodes issue a new incremental cost message, they set $H = 1$.

The intermediate nodes receiving the incremental cost message execute the following:

$$\begin{array}{lll} \text{if}(E \leq C + H \cdot d) & C = E, & H = 1, \\ \text{else} & H = H + 1, & \end{array} \tag{3}$$

where $d = L_{\text{payload}}/L_{\text{packet}}$. Recall that E denotes the additional cost (hop-count) from the source joining to the existing path tree to the current node.

In (3), $C + H \cdot d$ represents the net increase of power consumption from the source nodes to the current node when using the current interim aggregation point. If this value is greater than or equal to the value of E, the current node is more optimal than the interim aggregation point. In such a case, the current node substitutes for the interim aggregation point, so that it sets $C = E$ and $H = 1$. Otherwise, it increments the value of H by one.

Figure 1 shows the search procedure of the optimal aggregation point. Here we assume that the packet length is one and the payload length is 0.6.

The overhead of AGIT compared to the original GIT merely comprises the hop-count field to store H; it can be negligible.

3. Simulation

3.1 Model and Assumption

We implemented the original GIT and the AGIT on a self-developed event-driven simulator engine.

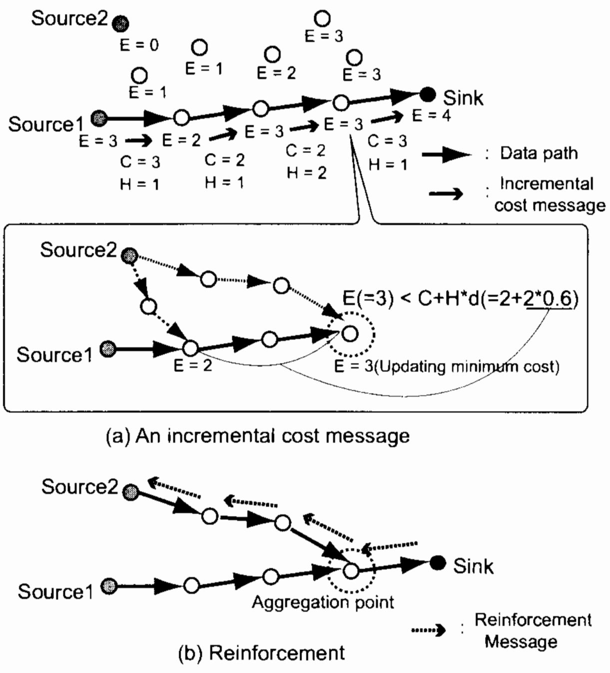

Fig. 1 An Example of Path Establishment in AGIT

In this simulator, 500 sensor nodes are deployed randomly in a 50×50 m^2 field. The transmission range is 5 m. One sink is located at $(45, 45)$ of the two-dimensional coordinate. The number of sources is varied from two to nine; they are arranged randomly in the field.

The packet has a 36-byte header. The payload length is varied as 4, 36, 108 and 216 bytes.

We implemented two schemes of the dissemination of the exploratory messages: "traditional exploratory", and "hop exploratory." The traditional exploratory scheme is the original one described in [7].

We implemented the ideal media access control (MAC) on our simulator, where no collisions occur.

Assuming the case by which the pass loss coefficient of $n = 2$, we modeled the energy consumption for transmission and reception of the packet of length l bits with distance R m, E_{tx} and E_{rx}, as follows:

$$E_{\mathrm{tx}} = (\alpha_{\mathrm{tx}} + \beta \times R^2) \times l, \quad (4)$$

$$E_{\mathrm{rx}} = \alpha_{\mathrm{rx}} \times l, \quad (5)$$

where α_{tx} and α_{rx} respectively denote the energy consumptions of the transmission circuit and the reception circuit, expressed as nanojoules per bit, and β denotes the radiation energy in appropriate units (nJ/bit/m^2) [4].

In simulation experiments, we use $\alpha_{\mathrm{tx}} = 50$ nJ/bit, $\alpha_{\mathrm{rx}} =$ 300 nJ/bit, and $\beta = 1.6$ nJ/bit/m^2.

3.2 Simulation Results

To evaluate how much the AGIT saves energy compared to the original GIT routing and opportunistic routing, we introduce the following metric, "*gain*", G:

$$G = \frac{\min(E_{\mathrm{GIT}}, E_{\mathrm{opp}}) - E_{\mathrm{AGIT}}}{\min(E_{\mathrm{GIT}}, E_{\mathrm{opp}})} \times 100, \quad (6)$$

where E_{GIT}, E_{AGIT}, and E_{opp} denote the energy consumption for data packet transmission on the path tree in the case of the GIT, the AGIT, and the opportunistic routing, respectively.

Figures 2 and 3 sshow the characteristics of gain as a function of the number of sources for different payload lengths.

Figure 2 shows the results of the traditional exploratory scheme. In this case, the exploratory messages are distributed network-wide. In the case of the small payload, 4 bytes, the gain is quite low because the aggregation ratio of the linear aggregation is almost identical that of the perfect aggregation. However, in the case of the medium payload, 36 bytes, which is the same as the header, the gain increases concomitant with the number of sources. This tendency is more remarkable in the case of the large payload, 108 bytes. However, in the case of too large payload, 216 bytes, the path tree will become similar to that of the opportunistic routing. Therefore, the gains decrease.

Figure 3 shows results of the hop exploratory scheme. From this figure, we can see that the AGIT routing is still more efficient than GIT routing and opportunistic routing, but the values of gain are decreased in comparison to those of the traditional exploratory scheme because the spread area of the exploratory messages is smaller than the traditional exploratory scheme.

The gain values are smaller than the expected values obtained from analysis. For analysis, we assume a dense network. However, in the simulation, the nodes are deployed in a discrete fashion. For that reason, the range of choices for the optimal aggregation point in the simulation is smaller than that for the analysis.

In Figs. 2 and 3, the gain fluctuates especially for the cases of 36-byte and 108-byte payloads, even though the average value for each case is led from 50 trials. This is because the variance of gain in this case is relatively large compared to the other cases.

Figures 2 and 3 show the gain in the data transmission phase. From this viewpoint, the traditional exploration is preferable. However, it includes the most overhead to construct the path tree. For that reason, we investigate the amount of the overhead. Figure 4 shows the total energy consumption of the entire network between the issue of the interest and the completion of the receptions of one data packet from every source. This figure indicates that the traditional exploratory scheme has more overhead than the others. A trade-off exists between the gain of the data transmission phase and the overhead of the path tree construction phase. The answer to the problem depends on the applications: more precisely, it depends on how long the data transmission phase lasts.

4. Conclusions

This paper presented the aggregation efficiency-aware GIT (AGIT), and also described analyses incorporating the suppression scheme for exploratory messages: hop exploratory.

The AGIT routing can construct a more efficient path tree than the original GIT. The improvement becomes more remarkable as the payload packet length becomes larger and/or more sources exist. Our simulation results demonstrate that the AGIT achieves up to 8% of the gain for the energy consumption of the data transmission compared to the original GIT. Our simulation results also emphasize that the suppression scheme, hop exploratory, reduces energy consumption up to 40%.

These suppression schemes might limit the flexibility of path findings or weaken the fault tolerance. Evaluation of those issues is left for future works.

Acknowledgment

This research work was partially supported by a Grant-in-Aid for Young Scientists (B), No. 16700066, 2005, from the

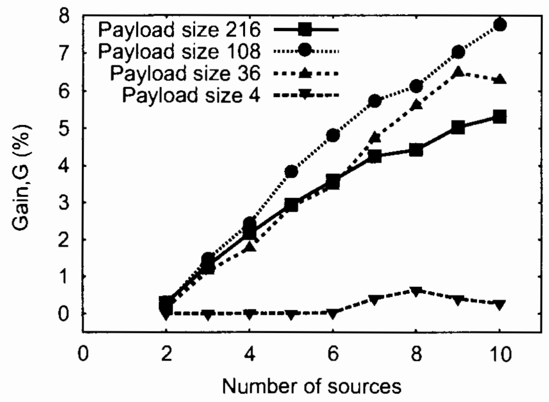

Fig. 2 Gains by nodes on the path tree in the case of the traditional exploratory scheme.

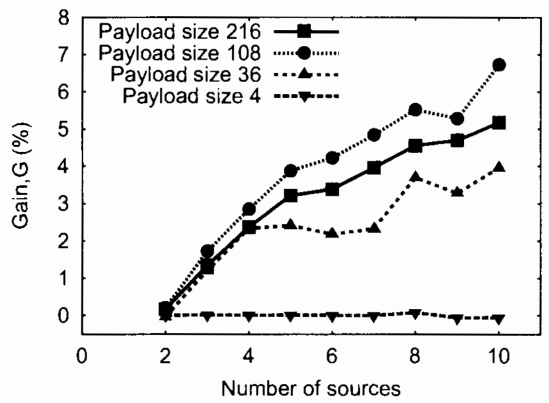

Fig. 3 Gains by nodes on the path tree in the case of the hop exploratory scheme.

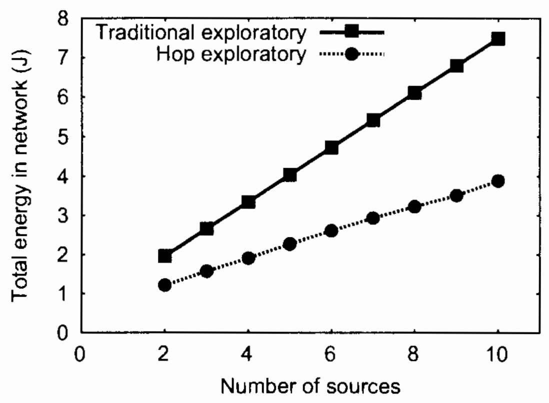

Fig. 4 Total energy consumption in whole network until the reception of the data packets from all sources.

Ministry of Education, Culture, Sports, Science and Technology, Japan, and by the Kayamori Foundation for Advancement of Information Science.

References

[1] T. F. Abdelzaher, T. He, and J. A. Stankovic, "Feedback Control of Data Aggregation in Sensor Networks," *IEEE Conference on Decision and Control*, Dec. 2004.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks (Elsevier) Journal*, vol.38, pp.393–422, March 2002.

[3] K. Akkaya, M. Younis, and M. Youssef, "Efficient Aggregation of Delay-Constrained Data in Wireless Sensor Networks," *Proc. of Internet Compatible QoS in Ad Hoc Wireless Networks 2005*, Jan. 2005.

[4] M. Bhardwaj, T. Garnett, A. P. Chandrakasan, "Upper Bounds on the Lifetime of Sensor Networks," *Proc. of ICC*, pp.785–790, June 2001.

[5] M. Enachescu, A. Goel, R. Govindan, and R. Motwani. "Scale Free Aggregation in Sensor Networks," *Proc. of the First International Workshop on Algorithmic Aspects of Wireless Sensor Networks*, pp.71–84, July 2004.

[6] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan, "Building Efficient Wireless Sensor Networks with Low-level Naming," *Proc. of the ACM Symposium on Operating Systems Principles*, Oct. 2001.

[7] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed Diffusion for Wireless Sensor Networking," *IEEE/ACM Transactions on Networking*, Feb. 2003.

[8] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann, "Impact of Density on Data Aggregation in Wireless Sensor Networks," *Proc. of the 22nd International Conference on Distributed Computing Systems*, Nov. 2001.

[9] M. Khan, G. Pandurangan and B. Bhargava, "Energy-Efficient Routing Schemes for Wireless Sensor Networks," *Tech. Rep. of Department of Computer Science, Purdue University*, CSD TR 03-013, July 2003.

[10] B. Krishnamachari, D. Estrinf, and S. Wicker, "Modelling Data-Centric Routing in Wireless Sensor Networks," *IEEE INFOCOM*, June 2002.

[11] B. Krishnamachari, D. Estrin, and S. Wicker, "The Impact of Data Aggregation in Wireless Sensor Networks," *Proc. of the 22nd International Conference on Distributed Computing Systems*, July 2002.

[12] D. Petrovic, C. Shah, K. Ramchandran, and J. Rabaey, "Data Funneling: Routing with Aggregation and Compression for Wireless Sensor Networks," *IEEE Sensor Network Protocols Applications, Anchorage*, May 2003.

[13] A. Wang, W. B. Heinzelman, A. Sinha, and A. P. Chandrakasan, "Energy-Scalable for Battery-Operated MicroSensor Networks," *Kluwer Journal of VLSI Signal Processing*, vol.29, pp.223–237, Nov. 2001.

[14] J. Zhao, R. Govindan, and D. Estrin, "Computing Aggregates for Monitoring Wireless Sensor Networks," Proc. of IEEE International Workshop on Sensor Network Protocols and Applications," May 2003.