Multi-Agent Itinerary Planning for Wireless Sensor Networks

Min Chen[1], Sergio Gonzalez[2], Yan Zhang[3], and Victor C.M. Leung[2]

[1] School of Computer Science & Engineering, Seoul National University, 151-744, Korea
`mchen@mmlab.snu.ac.kr`
[2] Elect & Comp Eng, University of British Columbia, V6T 1Z4, Canada
`sergiog,vleung@ece.ubc.ca`
[3] Simula Research Laboratory, 1325 Lysaker, Norway
`yanzhang@ieee.org`

Abstract. Agent-based data collection and aggregation have been proved to be efficient in wireless sensor networks (WSNs). While most of existing work focus on designing various single agent based itinerary planning (SIP) algorithms by considering energy-efficiency and/or aggregation efficiency, this paper identifies the drawbacks of this approach in large scale network, and proposes a solution through multi-agent based itinerary planning (MIP). A novel framework is presented to divide our MIP algorithm into four parts: visiting central location (VCL) selection algorithm, source-grouping algorithm, SIP algorithm and its iterative algorithm. Our simulation results have demonstrated that the proposed scheme lowers delay and improves the integrated energy-delay performance compared to the existing solutions with the similar computation complexity.

Keywords: Wireless sensor networks, mobile agent, itinerary planning.

1 Introduction

The application-specific nature of a wireless sensor network (WSN) requires that sensor nodes have various capabilities. It would be impractical to store all the programs needed in the local memory of embedded sensors to run every possible application, due to the tight memory constraints. The intrinsically flexible features of mobile agent (MA) make it adaptable to diverse network conditions in dynamically reconfigurable WSNs.

An agent deployed in a sensor network is a special kind of software that migrates among network nodes to carry out a task autonomously, in order to achieve the objectives of the sink node.

Compared to its traditional client/server computing communications mechanism counterpart, mobile agent based computing has exhibited its unique efficiency in context-aware sensory environments [1, 2, 3, 4, 5, 6, 7, 8]. In a previous survey [1], we separated the agent design process for WSNs into four parts: architecture, itinerary planning, middleware system design, and agent cooperation for the design, development, and deployment of MA systems for high-level inference and surveillance in WSNs.

Among the four components, itinerary planning determines the order of nodes to be visited during agent migration, which has a significant impact on energy performance

N. Bartolini et al. (Eds.): QShine/AAA-IDEA 2009, LNICST 22, pp. 584–597, 2009.

of the MA systems. Though the agent itinerary is critical to the network performance, it has been shown that finding an optimal itinerary is NP hard and still an open area of research. Therefore, heuristic algorithms [2, 4, 10] and genetic algorithms [5] are generally used to compute itineraries with a sub-optimal performance. Though our previously introduced IEMF and IEMA approaches [10] exhibit higher performance in terms of energy efficiency and delay compared to the existing solutions, the limitation of utilizing a single agent to perform the whole task, making the algorithm unscalable in applications with a large number of source nodes needed to be visited. Typically, single agent itinerary planning algorithms have high efficiency in the applications with the following characteristics:

- The source nodes are distributed geographically close to each other.
- The number of source nodes is not large.

For a large scale sensor networks, with many nodes to be visited, single agent data dissemination exhibits the following pitfalls:

1. *Large Delay*: Extensive delay is needed when a single agent works for networks comprising hundreds of sensor nodes.
2. *Unbalanced load*: There are two kinds of unbalancing problems while using a single agent. First, in the perspective of the whole network, all of the traffic load is put on a single flow. Therefore, sensor nodes in the agent itinerary will deplete energy quickly than other nodes. Secondly, from the perspective of the itinerary, the agent size increases continuously while it visits source nodes, and so the agent transmissions will consume more energy in its itinerary back to the sink node.
3. *Insecurity with large accumulated size*: The increasing amount of data accumulated by the agent during its migration task increases its chances of being lost due to noise in the wireless medium. Thus, the longer the itinerary, the higher risky of the agent-based migration becomes.

In this paper, we propose a novel Multi-agent Itinerary Planning (MIP) algorithm to address the above issue. Traditionally, Single-agent Itinerary Planning (SIP) includes the following two challenges:

- Selecting the set of the source nodes to be visited by the mobile agent.
- Determining a node visiting sequence in an energy-efficient manner.

Compared to existing SIP proposals, the main contributions of this paper are listed as follows:

- We introduce a novel source-grouping algorithm. Note in [11], clustering based architecture is utilized to facilitate mobile agent based data dissemination. Though our source-grouping algorithm partitions source nodes into several sets, which has a similar effect of grouping source nodes in clusters, we do not set up a hierarchical structure. Thus, our algorithm does not have any control message overhead for the clustering process.
- We propose an iterative algorithm for MIP solution.
- We propose a generic framework to design a MIP algorithm. Within this framework, any SIP algorithm can be extended to the corresponding MIP algorithm, where the SIP algorithm will be carried out iteratively until the source list is empty.

The remainder of the paper is organized as follows. The problem is stated in Section 2. We present the proposed MIP algorithm in Section 3. Our simulation studies are reported in Section 4. Section 5 concludes the paper.

2 Problem Statement

2.1 Motivation

In this section, the motivation for MIP proposal is illustrated through Eqns.(1) and (2). The agent size at the kth source depends on three parts: (1) the initial agent size (l_{ma}^0), which includes size of processing code and agent header; (2) size of reduced payload when visiting the first source node ($l_{data} \cdot (1 - r_1)$), where r_1 is the data reduction ratio at the first source. Note that there is no data aggregation at the first source; (3) accumulated size of the aggregated data payload after local processing from the second source node to the present source ($\sum_{i=2}^{k} l_{data} \cdot (1 - r_i) \cdot (1 - \rho_i)$)[1]. Thus, the final agent size increases linearly with the source number, as shown in Eqn.(1).

$$\begin{aligned} l_{ma}^k = l_{ma}^0 &+ l_{data} \cdot (1 - r_1) \\ &+ \sum_{i=2}^{k} l_{data} \cdot (1 - r_i) \cdot (1 - \rho_i). \end{aligned} \tag{1}$$

$$E_{itinerary} = E_0^1 + \sum_{k=2}^{n} E_{k-1}^{k}(l_{ma}^{k-1}) + E_n^0. \tag{2}$$

Table 1. Notation

Symbol	Definition
l_{data}	the size of raw sensory data at a source node.
l_{ma}^0	the size of mobile agent when dispatched from the sink.
r_i	the reduction ratio at the kth source by agent assisted local processing.
ρ_i	aggregation ratio at the kth source by agent for data redundancy elimination.
l_{ma}^k	the agent size when it leaves the kth source.
N	the number of source nodes needed to be visited.
$E_{k-1}^k(l_{ma})$	the communication energy cost during a mobile agent roams from source *k-1* to source k with agent size l_{ma}.

Fig. 1 presents a typical scenario of single agent based data dissemination. In order to calculate the itinerary cost ($E_{itinerary}$), we divide the whole itinerary cost into three parts: (1) from the sink node to the first source node S_1, only the processing code and

---

[1] Please refer Table 1 for the definitions of ρ_i and l_{data}.

agent header are included in the MA packet. We denote the communication energy consumption in this part by E_0^1; (2) the second part starting from the time when MA leaves the first source node to the time when it visits the last source node S_n. The communication energy consumption in this phase is denoted by $\sum_{k=2}^{n} E_{k-1}^{k}(l_{ma}^{k-1})$, where $E_{k-1}^{k}(l_{ma}^{k-1})$ represent the communication energy cost for the MA to roam from source *k-1* to source k with agent size l_{ma}; (3) the third part starting from the time when MA finishes visiting all the source nodes to the time when it returns to the sink. The communication energy consumption in this part is denoted by E_n^0. Eqn.(2) shows that the itinerary cost is a squarely increasing function of the source node number, which causes the performance of the SIP algorithm to deteriorate in large scale sensor networks. The end-to-end agent delay exhibits a trend that is congruent to the similar trend as the itinerary cost. Thus, we are motivated to design a MIP algorithm that possesses the flexibility of adapting to the specific network parameters, such as network size, source node number, reduction ratio, aggregation ratio, sensor data size, etc. Specifically, a SIP algorithm can be deemed as a particular output of the MIP algorithm with a single agent.

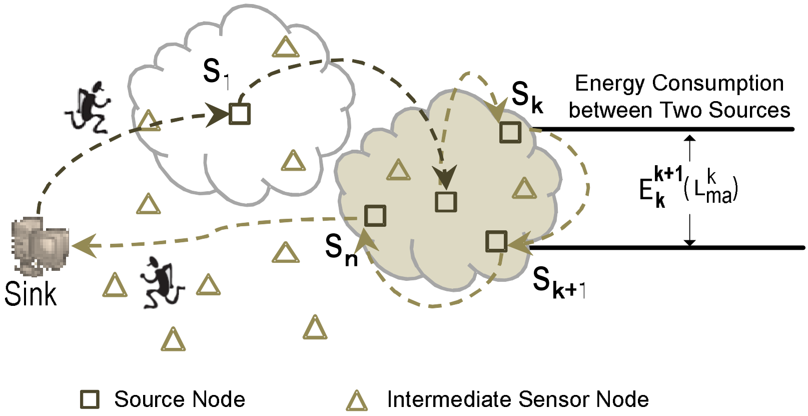

Fig. 1. Illustration of Single Agent based Itinerary Planning

2.2 A Generic Multi-Agent Itinerary Planning Algorithm

We state our assumptions and define a generic MIP algorithm in this section as follows:

- primary itinerary design algorithms are executed at the sink, which has relatively plenty of resources in terms of energy and computation.[2]
- the sink node knows the geographic information of all the source nodes. Note that in our algorithm, only source locations are needed, while the other algorithms [4, 5] need all of the nodes' geographical positions.

In fact, the above assumptions are common in most of the solutions presented in [4, 5, 10] for the SIP problem. The previous SIP algorithms assume that the set of source

---

[2] MAs may deal with unexpected failures of arriving next source nodes, as soon as failure is detected, MA change the source destination node scheduled to be visited after the failed node.

nodes to visit is predetermined. In contrast, our MIP algorithm needs to group source nodes for different mobile agents, since the determination of source visiting set is a dynamic process. The proposed MIP algorithm can be deemed as the iterative version of a SIP solution, which can be divided into four parts:

- *Selection of Visiting Central Location (VCL) for an agent*: While using multiple agents, it is a challenging issue to use the least number of them while achieving the required coverage of source nodes. Strategically, the agent's VCL is selected to the center of area with a high source node density. Finding an optimal agent number is also a NP-hard hard problem.
- *Determining the source visiting set*: In order to determining the source visiting set, we first isolate the visiting area, which is typically a circle/oval centered at the VCL and it has a certain radius. All of the source nodes in the disk will be included in the visiting list of the agent.
- *Determining a source-visiting sequence*: This is the itinerary plan for the current agent. In this step, the problem is simplified into the *Single-agent Itinerary Planning* problem, whereby existing SIP solutions can be applied, such as LCF, GCF, MADD, IEMF and IEMA, etc.
- *Algorithm iteration*: If there are source uncovered source nodes, the next VCL will be calculated based on the remaining set of source nodes. The previous process will repeat until all of the source nodes have been assigned to a mobile agent.

3 Proposed MIP Algorithm

VCL-Selection Algorithm. The basic idea of the proposed *visiting central location* (VCL) selection algorithm is to distribute each source's impact factor to other source nodes. Let n denote the source number. Then, each source will receive $n - 1$ impact factors from other source nodes, and one from itself. After calculating the accumulated impact factor, the location of the source with the largest accumulated impact factor will be selected as VCL.

We achieve this by using the analogy of a gravity field: a source node is modeled by a small iron ball, and the network is seen as an elastic plane, as shown in Fig. 2(a). When the iron ball is put on the elastic plane, the plane will be naturally distorted to the shape as shown in Fig. 2(b). In our approach, we map the physical model to a sensor network in a way where each source will contribute with a certain gravity impact to a fixed location. If we overlap all of the source nodes' gravity fields, there must be a location suffering the largest gravity[3]. We define this location as VCL.

However, there are unlimited locations in the plane. Therefore, in order to reduce computational complexity, we make the following simplifications:

- the gravity impact is quantized by hop count between two source nodes.
- only the location of a source node is considered as a candidate to be selected as VCL.

---

[3] In Physics, this is analogous to a Boltzmann Machine, and gradient descent.

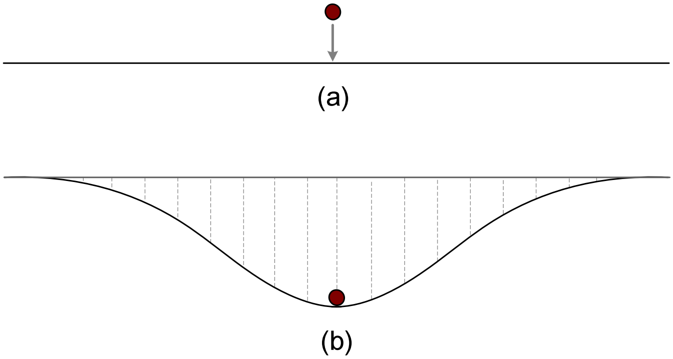

Fig. 2. Analog to Illustration of Calculating the Impact Factor between Two Source Nodes

We denote the set of n nodes by V_n. For any two source nodes $i, j \in V_n$, d_{ij} denotes the distance between i and j. Then, we can estimate the hop count between i and j as $H_i^j = \lceil \frac{d(k-1,k)}{R} \rceil$, where R represents the maximum transmission range. To approximate the effect of a real gravity field, a gauss function is adopted to calculate the impact factor between i and j:

$$G_{ij} = e^{-\frac{(H_i^j - 1)^2}{2\sigma^2}}. \tag{3}$$

Fig. 3 shows an example with σ set to 8. A suitable setting should be heuristically selected for different network scale and different requirements of grouping effect.

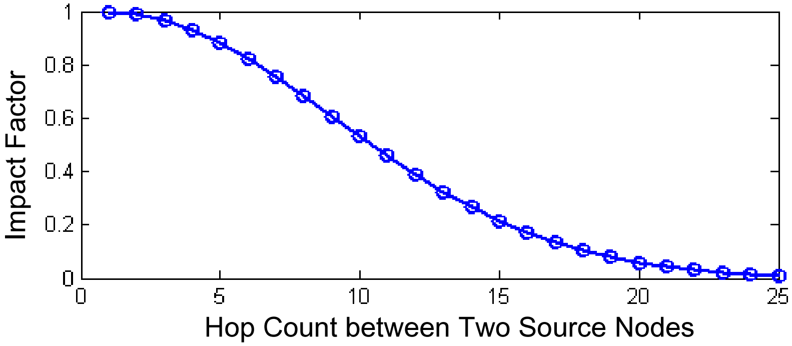

Fig. 3. Illustration of Calculating the Impact Factor between Two Source Nodes

The pseudo code of the VCL-selection algorithm is listed at Algorithm 1.

Source-grouping algorithm. Our source-grouping algorithm is very simple. Let $A(VCL, R)$ denote the circular area centered at VCL with a radius of R. Then, all of the source nodes within $A(VCL, R)$ will be included in the visiting list which is assigned to the current agent. Algorithm 2 shows the pseudo-code of the proposed source-grouping algorithm.

Algorithm 1. VCL-selection algorithm for the set of source nodes (V_m)

```
for each source $i$ in $V_m$ do
    $G_i \leftarrow 0$;
end for
for each source $i$ in $V_m$ do
    for each source $j$ in $V_m$ do
        calculate $G_{ij}$ according to Eqn.(3);
        $G_i \leftarrow G_i + G_{ij}$;
    end for
end for
for each source $k$ in $V_m$ do
    if $G_k = min\{G_i| \in V_m\}$ then
        select the position of node $k$ as VCL;
        break;
    end if
end for
```

Algorithm 2. Source-grouping algorithm for the set of source nodes (V_m)

```
for each source $i$ in $V_m$ do
      calculate the distance ($d_{vcl,i}$) between VCL and node $i$;
    if $d_{vcl,i} < R$ then
        $V_{left} \leftarrow V_m - i$;
        $V_{group} \leftarrow V_{group}$ + $i$;
    end if
end for
```

Iteration based MIP Algorithm. For each iteration, a new VCL will be calculated for the remaining source list. Then, a new list of source nodes will be assigned to a mobile agent. To this moment, the itinerary for the agent can be planned by any SIP algorithms. In this paper, some typical SIP algorithms are tested, such as LCF, GCF and IEMF. If the remaining source list is not empty, the above process will repeat until all of the source nodes have been assigned to a mobile agent. The pseudo code of the iteration based MIP algorithm is shown at Algorithm 3.

Algorithm 3. MIP algorithm for the whole set of source nodes (V_n)

```
  $V_{left} \leftarrow V_n$;
loop
    if $V_{left}$ is not empty then
        calculate VCL $V_{left}$ according to Algorithm 1;
        calculate $V_{group}$ according to Algorithm 2;
        perform SIP algorithm for $V_{group}$;
        updated $V_{left}$ according to Algorithm 2;
    end if
end loop
```

The computational complexity of Algorithms 1,2 and 3 is $O(n^2)$, and the one for our MIP scheme depends on the SIP algorithm. For example, if a SIP (e.g., LCF) has a computational complexity of $O(n^2)$, then a LCF-based MIP algorithm will have the same computation complexity.

4 Performance Evaluation

4.1 Simulation Setting

We implement the proposed MIP algorithm as well as the three existing SIP algorithms (LCF, GCF and IEMF) using OPNET Modeler, and perform extensive simulations. We choose a network where nodes are uniformly deployed within a 1000m $\times$ 500m field. To verify the scaling property of our algorithms, we select a large-scale network with 800 nodes. We assume that the sink node is located at the right side of the field and multiple source nodes are randomly distributed in the network.

The sensor application module consists of a constant-bit-rate source, which generates a sensora data report every 1 s (1024 bits each). As in [10], we use IEEE 802.11 DCF as the underlying MAC, and the radio transmission range is set to 60 m. The data rate of the wireless channel is 1 Mb/s. All messages are 64 bits in length. For consistency, we use the same energy consumption model as in [9]. The initial energy of each node is 5 Joules. The power consumptions for transmission, reception and idling are 0.66 W, 0.395 W, and 0.035 W, respectively. We count for all types of energy consumptions in the simulations, including transmission, recption, idling, overhearing, collisions and other unsuccessful transmissions, MAC layer headers, retransmissions, and RTS/CTS/ACKs.

We consider the following four performance metrics:

- *Task Duration*: in a SIP algorithm, it is the average delay from the time when a MA is dispatched by the sink to the time when the agent returns to the sink. In our MIP algorithm, since multiple agents work in parallel, there must be one agent which returns to the sink at last. Then, the task duration of our MIP algorithm is the delay of that agent.
- *Average Communication Energy*: the total communication energy consumption, including transmitting, receiving, retransmissions, overhearing and collision, over the total number of distinct reports received at the sink.
- *Hop Count*: in SIP, it is the average hop count of a mobile agent itinerary. In MIP, it is the accumulated hop counts of all the agents.
- *Integrated Performance*: For time-sensitive applications over energy constrained WSNs, we consider both delay and energy performances, and evaluate the integrated performance (denoted by η) in terms of task duration and average communication energy. The smaller the value of η is, the better the integrated performance will be.

$$\eta = energy \cdot delay. \tag{4}$$

In all the figures presented in this section, each data point is the average of 25 simulation, which runs with different random seeds. The mobile agent specific parameters are shown in Table 2.

Table 2. Simulation Parameters

Raw Data Reduction Ratio (r)	0.8
Aggregation Ratio (ρ)	0.9
MA Accessing Delay (τ)	10 ms
Data Processing Rate (V_p)	50 Mbps
Size of Sensed (Raw) Data (l_{data})	Default: 2048 bits
Size of Processing Code (l_{proc})	1024 bits
The Number of Source Node (n)	Default: 40
Radius to the center point	Default: 255

4.2 Simulation Results

Multiple Itineraries Construction in the Proposed MIP Algorithm. In this section, we will show the snapshot of OPNET simulation for MIP algorithm, and a typical SIP algorithm (i.e., LCF). Fig. 4 shows the result of source-grouping and itinerary planning for the first mobile agent. The circled node is selected as the first VCL, since it has the highest accumulated impact factor of 14.87. The second VCL has the highest impact factor of 4.14, which is much smaller than that of the first one. It is because the number of candidate source nodes is smaller than that of the first round selection, as shown in Fig. 5. Note that the number of left source nodes will become smaller and smaller, and only three source nodes are visited by the third agent, as shown in Fig. 6. Since the three agents are dispatched by the sink node in parallel, they collect sensory data concurrently. Observed from above figures, the third agent will return to the sink first, then the second agent. Finally, the task duration is actually the delay of the first agent. In order to compare MIP and SIP algorithms, the planned itinerary by LCF is plotted in Fig. 7. Intuitively, the length of itinerary is longer than that of the first agent in

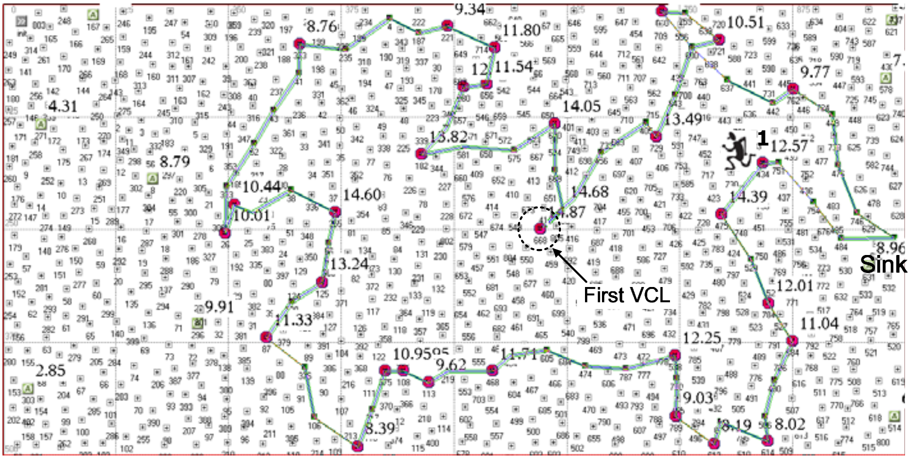

Fig. 4. The snapshot of the first itinerary in MIP algorithm

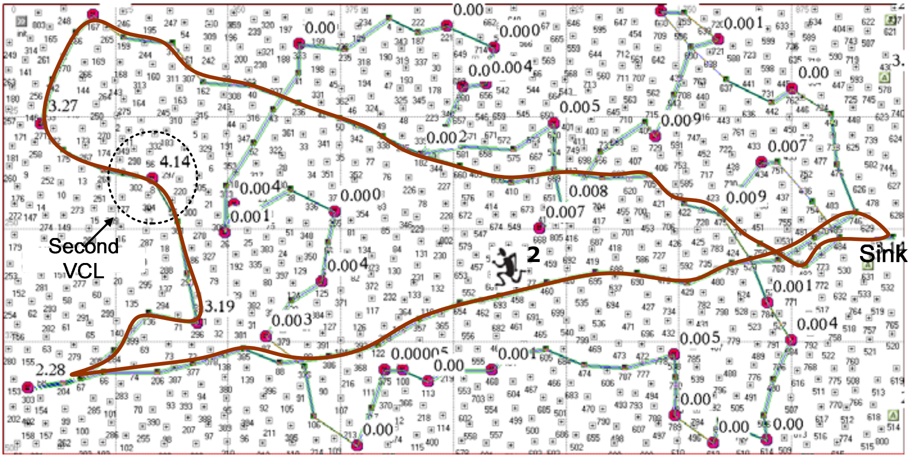

Fig. 5. The snapshot of the second itinerary in MIP algorithm

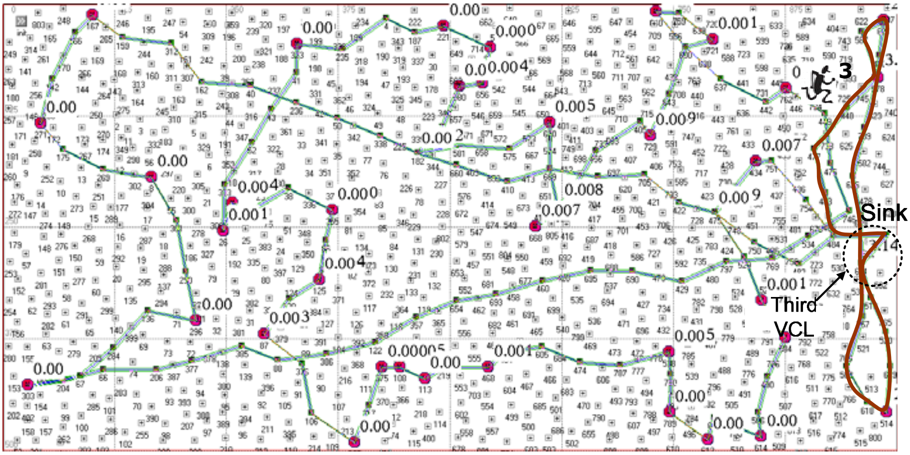

Fig. 6. The snapshot of the third itinerary in MIP algorithm

MIP algorithm. Thus, MIP algorithm will yield a shorter task duration than the SIP algorithm.

Performance Comparison of *MIP*, *LCF*, *GCF*, *MADD*, *IEMF*, *IEMF* and *IEMA* with Varying Source Number. In this section, MIP algorithm is compared to most of existing SIP algorithms, such as LCF, GCF, MADD, IEMF and IEMA, among which IEMA outstands other SIP algorithms [10].

A series of experimental networks of different source node number are created. For the same source number at different simulation seed, the distribution of the source nodes are recalculated through random selection.

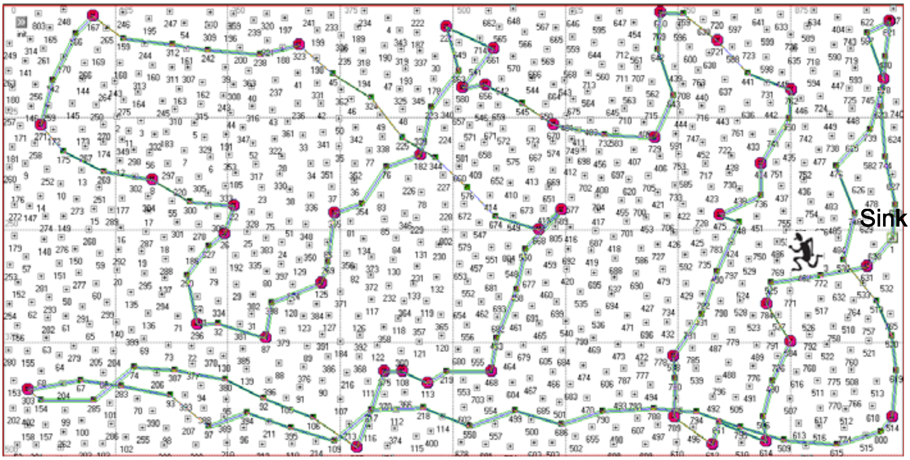

Fig. 7. The snapshot of LCF algorithm

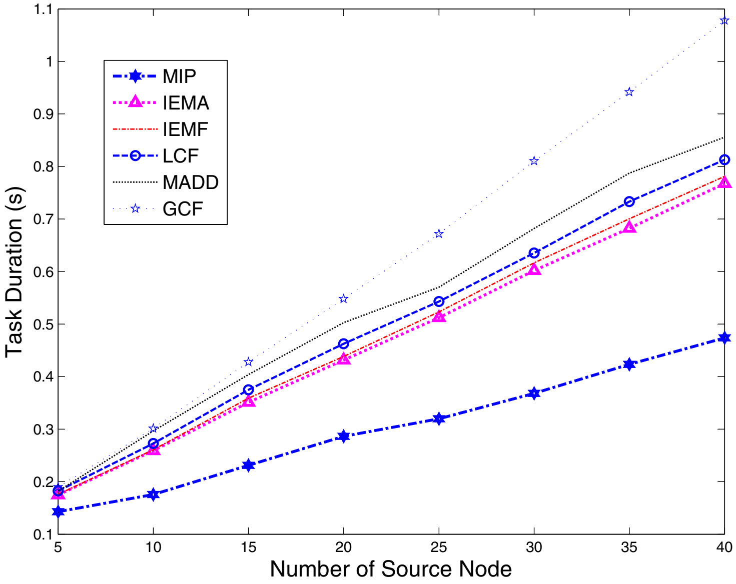

Fig. 8. Task durations

As shown in Fig. 8, MIP algorithm has absolute advantage in terms of task duration, which is only half of that of LCF. Note that the task duration of MIP is calculated fairly with SIP algorithms. Since we dispatch all of the mobile agents simultaneously, contention exists when multiple agents are close to each other. Even so, MIP still has superior delay performance than SIP algorithms.

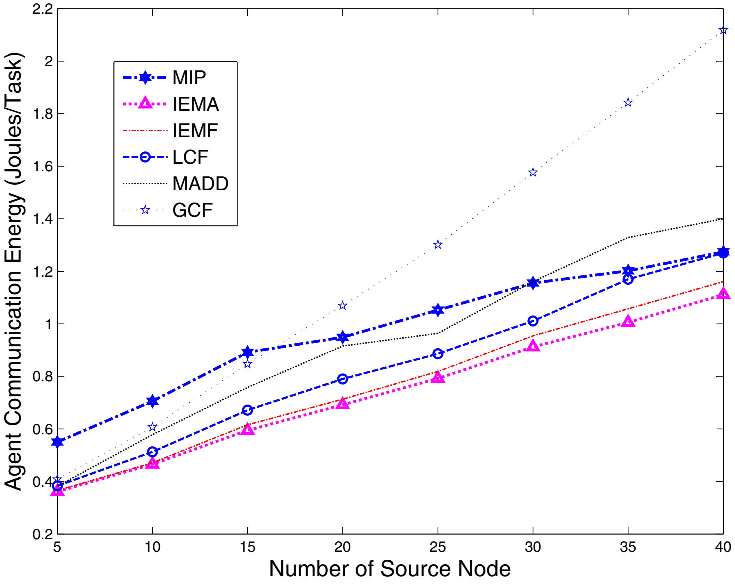

Fig. 9. Task communication energy

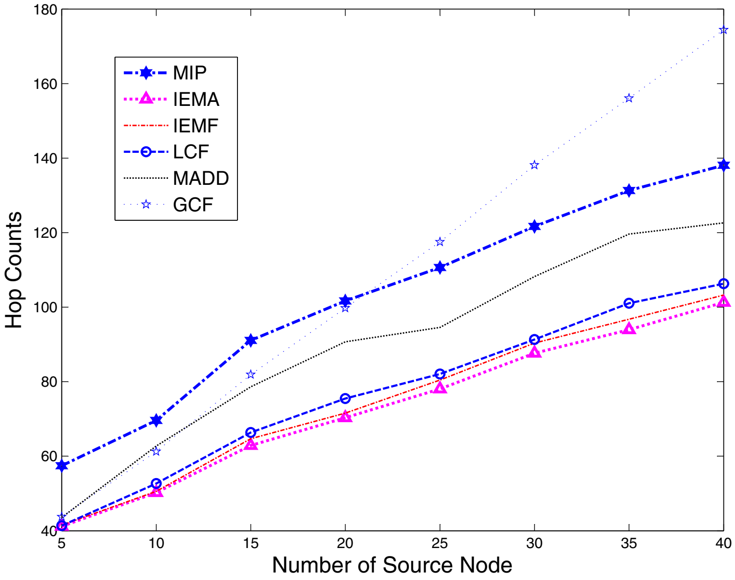

Fig. 10. Hop counts

In Fig. 9, the energy consumption of MIP algorithm is much higher than that of SIP algorithms when source number is small. Actually, it is only necessary for the usage of

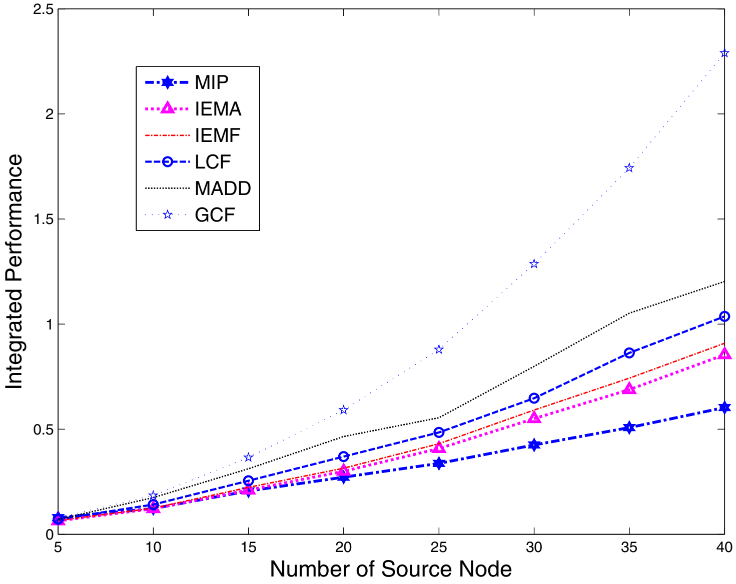

Fig. 11. Agent accumulated delay

multi-agent when source number is large. When the source number is 40, the energy consumption of MIP algorithm becomes comparable to those of SIP algorithms.

In Fig. 10, the accumulated hop counts of MIP algorithm is larger than the hop count of SIP. By comparison, it is important to consider the joint delay and energy performance, especially for delay constraint traffic in wireless sensor network, such as wireless multimedia sensor network, and video sensor networks [12]. Fig. 11 shows that MIP algorithm has the best integrated performance, which verifies effectiveness of the proposed algorithm.

5 Conclusions

In this paper, we addressed the problem of itinerary planning for multi-agent based data dissemination, facilitating concurrent sensory data collection to reduce task duration extensively. The proposed multi-agent itinerary planning (MIP) algorithm has the similar complexity with most of single agent based itinerary (SIP) algorithm, and can be flexibly adaptive to network dynamics in various network scales. We will propose more efficient source-grouping algorithm in our future work.

Acknowledgement. This work was supported in part by the Canadian Natural Sciences and Engineering Research Council under grant STPGP 322208-05. This work was supported in part by the IT R&D program of MKE/IITA [2007-F-038-03, Fundamental Technologies for the Future Internet] and grant number IITA-2009-C1090-0902-0006. OPNET University Program and The ICT at Seoul National University provides research facilities for this study.

References

1. Chen, M., Gonzalez, S., Leung, V.C.M.: Applications and design issues of mobile agents in wireless sensor networks. IEEE Wireless Communications 14(6), 20–26 (2007)
2. Chen, M., Kwon, T., Yuan, Y., Choi, Y., Leung, V.: Mobile agent-based directed diffusion in wireless sensor networks. EURASIP Journal on Advances in Signal Processing 2007(1), 219–242 (2007)
3. Chen, M., Kwon, T., Yuan, Y., Leung, V.C.M.: Mobile Agent Based Wireless Sensor Networks. Journal of Computers 1(1), 14–21 (2006)
4. Qi, H., Wang, F.: Optimal itinerary analysis for mobile agents in ad hoc wireless sensor networks. In: Proc. IEEE ICC 2001, Helsinki, Finland (2001)
5. Wu, Q., Iyengar, S.S., Rao, N.S.V., Barhen, J., Vaishnavi, V.K., Qi, H., Chakrabarty, K.: On computing mobile agent routes for data fusion in distributed sensor networks. IEEE Trans. Knowledge and Data Engineering 16(6), 740–753 (2004)
6. Qi, H., Xu, Y., Wang, X.: Mobile-Agent-Based Collaborative Signal and Information Processing in Sensor Networks. Proceedings of the IEEE 91(8), 1172–1183 (2003)
7. Xu, Y., Qi, H.: Mobile agent migration modeling and design for target tracking in wireless sensor networks. Ad Hoc Networks Journal 6(1), 1–16 (2008)
8. Tseng, Y., Kuo, S., Lee, H., Huang, C.: Location tracking in a wireless sensor network by mobile agents and its data fusion strategies. The Computer Journal 47(4), 448–460 (2004)
9. Mao, S., Hou, Y.T.: BeamStar: An edge-based approach to routing in wireless sensor networks. IEEE Trans. Mobile Computing 6(11), 1284–1296 (2007)
10. Chen, M., Leung, V., Mao, S., Kwon, T., Li, M.: Energy-efficient Itinerary Planning for Mobile Agents in Wireless Sensor Networks. In: IEEE International Conference on Communications (ICC 2009), Dresden, Germany (2009)
11. Xu, Y., Qi, H.: Distributed Computing Paradigms for Collaborative Signal and Information Processing in Sensor Networks. International Journal of Parallel and Distributed Computing 64(8), 945–959 (2004)
12. Akyildiz, I.F., Melodia, T., Chowdhury, K.R.: A Survery on Wireless Multimedia Sensor Networks. Computer Networks 51(4), 921–960 (2007)
13. `http://en.wikipedia.org/wiki/Intrinsic_safety`
14. `http://www.msha.gov/techsupp/PEDLocating/MSHAApprovedPEDproducts.pdf`
15. Liu, H., Lai, X., Ma, D.: Application of CDMA Technology based Communication in Underground Coal Mines. Mobile Communication (China) 29(10), 113–114 (2005)
16. Chen, M., Gonzalez, S., Zhang, Q., Li, M., Leung, V.: 2G-RFID based E-healthcare System. In: IEEE Wireless Communications Magazine, Special Issue on Wireless Technologies for E-healthcare (to appear)