International Journal of Innovative
Computing, Information and Control
Volume 8, Number 5(A), May 2012

ICIC International ©2012 ISSN 1349-4198
pp. 3029–3047

DESIGN AND ANALYSIS OF LOCATION CONTROL FOR MOBILE AGENT IN MOBILE INFORMATION SYSTEMS

Ding-Chau Wang[1,*] and Chao-Chun Chen[2]

[1]Department of Information Management
[2]Department of Computer Science and Information Engineering
Southern Taiwan University
No. 1, Nan-Tai St., Yong-Kang Dist., Tainan 710, Taiwan
*Corresponding author: zh9@mail.stut.edu.tw; chencc@mail.stut.edu.tw

Received January 2011; revised May 2011

Abstract. *Retrieval of data using mobile agents has gained popularity, due to the advancements in mobile computing techniques. However, the transmission of wireless communication is still unreliable and bandwidth-limited. This paper investigates this issue and tries to find the most suitable location for a mobile agent in a vast network. We first propose a mechanism and operations in support of this strategy. Then, we develop an analysis model to estimate the optimal location for a mobile agent. Our experimental results show that at an optimal location, about* 27% ∼ 116% *of communication cost can be saved.*
Keywords: Mobile computing, Distributed computation, Wireless communication, Data management, Agent

1. **Introduction.** Mobile computing is now a reality owing to the rapid advances in wireless communication and electronic technologies. The proposed mobility mechanisms [1, 18, 22] can help mobile users (MUs) to access data seamlessly, while they are roaming. Very often MUs need to interact with applications of very large and complex systems, such as mobile commerce applications [14], transcodes of wireless multimedia data, mobile robots [24] and database management systems (DBMS) [8]. When executing a task like a database transaction, a MU needs to frequently communicate with such a remote server. However, the transmission of wireless communication is still unreliable (e.g., frequent disconnection or data lost) and bandwidth-limited. A large amount of waiting time would be incurred if the MU directly communicates with remote peers, when required. Therefore, an important issue in this environment is to reduce the utilization of wireless communication. A software approach to overcome the above problem is to place a special-purposed mobile agent (MA) [4, 5, 10, 11, 12, 13, 26] in the wired network. This MA executes communication-bound tasks for the MU, on receiving a request message. Consequently, the MU only needs to connect to the wireless network twice, while executing a communication-bound task. The first time is to send the request to the agent and the second time is to receive the request result from the agent. The advantages of having a MA include:

- *Reducing the utilization of wireless communication.* In many tasks, the size of intermediate data is greater than that of the result data. These intermediate data need to be transferred to the MA. Wireless communication between the MU and the MA is not needed before the MA sends results back to the MU, which helps to relieve the possible bottleneck caused by wireless communication.

- *Reducing workload of mobile devices.* Mobile devices are equipped with low-end hardware making it difficult to deal with complex tasks. A MA can take over those tasks and deal with them with the help of the fixed server, consequently reducing the workload of the mobile device.
- *Lowering the risks of disconnection.* In traditional client/server model, when disconnection occurs, the executing task is suspended or aborted. The use of a MA helps to avoid these. A MA will keep the result data until a MU reconnects to the network.

In concept, a MA plays the role of surrogate of a MU in wired network. Suppose a MU and a fixed server are at the two ends of a communication link, and the MA is placed somewhere in between these two ends. Intuitively, the MA should be placed close to the MU because its purpose is to serve the MU [3, 9, 27]. Therefore, whenever the MU moves, it moves along. This approach is unsuitable when the MU is highly mobile, as too much movement of the MA incurs overhead. Therefore, in the past, most designs placed their agents close to the other end, i.e., the fixed server, to serve the MU [19]. This extreme, however, may incur a high overhead when a large result needs to be transmitted through several hops to the MU. In order to avoid the weaknesses of the above two extremes, we designed a MA controlling mechanism and examined the most suitable location for the MA. We developed an analysis model to represent and evaluate the effect of the distance between the MA and the MU. Our result shows that the mobility of the MU indeed has a significant influence on where the MA is to be placed. In summary, the contributions of this paper is three-fold:

- First, we design a MA controlling mechanism and describe how the system is managed. We formulate the networks as hierarchical structure because the distance between a MA and a MU is easily obtained in such network structure.
- Second, we develop the analysis model for performance evaluation.
- Third, our experimental results indicate that in most cases, placing the MA at the end of the wired network cannot produce optimal performance, which is contradictory to the past unproved belief. The experimental results also show that poorer the communication environment (e.g., highway, crowded city, or area of slow and unreliable networks), the more important the MA's location becomes.

The remainder of this paper is organised as follows. In Section 2, we present system management. The system analysis is developed in Section 3. Performance evaluation is presented in Section 4, while the issue of an optimal location for the MA is studied in Section 5. Then, we discuss the related implementation issues in Section 6. Finally, the paper is summarised in Section 7.

2. **System Management.**

In this section, we first introduce the architecture of wired and wireless networks. Then, we describe how a MA interacts with a MU.

2.1. **Network architecture.**

Figure 1 shows the system architecture on which our mechanism is designed. Geographically, the regions are divided into cells, whose sizes range from 50 metres to tens of kilometres in diameter, depending upon the number of possible MUs within a cell. Each cell has a *base station* (BS), and the management of the locations of MUs within a cell and the connection of these MUs with the outside world are all through the *mobile support station* (MSS) of the cell [15, 22]. The BS and MSS are usually located nearly and regarded as the same one in the literature.

In the wired network, the fixed servers offering working places for MA's are called *MA nodes*. MA nodes are connected to each other through wired networks having different

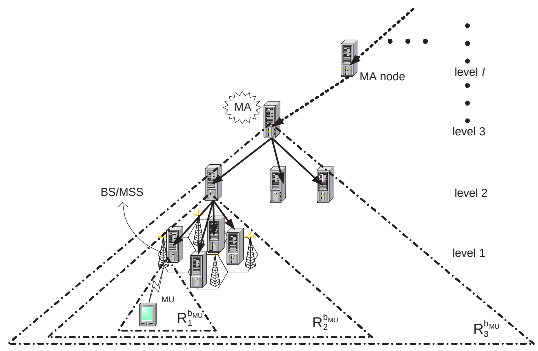


FIGURE 1. The network architecture of the mobile environment

bandwidth, and the bandwidth of two connected MA nodes determines the distance between them. The connected MA nodes form an arbitrary topology. For ease of presenting our idea and without the loss of generality, we assume the MA nodes form a hierarchy topology over the wired network [1]. All BSs are inherently the leaf nodes. Level 1 of the hierarchy is at the leaves of the tree, and level 2 nodes are the parents of the level 1 nodes, and so on. Theoretically, the hierarchy can extend to an infinite number of levels. For simplification, we also make two assumptions for the wired network: (1) the BS and the corresponding level 2 MA node can be viewed as the same node; and (2) a MA node is also a router.

As mentioned above, a cell is a small geographical area overseen by a base station. Let the BS that a MU resides be b_{MU}. The ancestor MA node of b_{MU}, therefore, oversees a large region formed by the small areas of the sibling BSs of b_{MU}, who have this same ancestor. If the ancestor is at level ℓ of the hierarchy, we denote its overseen region as $R_\ell^{b_{MU}}$ for convenience of presentation.

A MA consists of the following two components:

- MA process: contains codes and MU's profile. The codes determine functions of the MA, e.g., communication interface, mobility modules and request processing modules. The MU's profile contains information on how to interact with the MU, e.g., personal information, moving behavior or history and MU's current location.
- Data repository: contains MU's interesting and frequently used data. The size of the data repository should be managed carefully, because too small a data repository does not help in saving required data, while too large a data repository could result in high communication cost when the MA moves.

2.2. **MA controlling mechanism.** As our primary objective is to find the best node of the MA hierarchy for placing a MA, we do not wish to blur our focus by introducing a highly complicated MA controlling mechanism. Therefore, a realistic and easy-to-implement MA controlling mechanism is proposed. The MA controlling mechanism contains two operations, *request operation* and *movement operation*. The request operation describes how the MA deals with the MU's request. The movement operation describes when and how the MA moves. The basic idea of the MA controlling mechanism is that *a served MU is always under the "coverage" of a MA*. The algorithms of the two operations will be formally presented in the following subsections.

In order to communicate with each other, the MU and the MA, both, contain an address. In the MU, the address is the MA's location and, therefore, called *MA_address*. Similarly, in the MA, the address is the MU's location and, therefore, called *MU_address*.

2.2.1. *Request operation.* First, we present the request operation of the MA controlling mechanism. A request operation is issued when a MU requests a MA to execute a task. The network communication in a request operation involves: (1) delivering the request from the MU to the MA; and (2) sending the result back from the MA to the MU. The detailed steps are shown in Algorithm 1.

Algorithm 1 Request Operation

1: The MU sends the request to b_{MU} (i.e., the MU currently residing base station). The location of the MA (i.e., the content of *MA_address*) is also included in the request.
2: b_{MU} extracts the location of the MA from the request, and delivers the request to the MA.
3: The MA processes the request and produces the result.
4: The MA sends the request to b_{MU} through the wired network.
5: b_{MU} forwards the result to the MU through the wireless network.

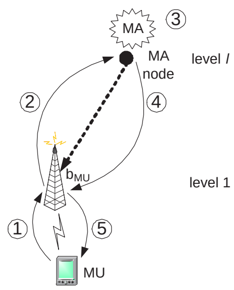


FIGURE 2. An example of the MA's request operation

Figure 2 illustrates the scenario. The dotted line between the BS and MA node in the figure represents the wired network, and the solid lines represent the communication between the MU and the MA. In addition, for convenience, figures are provided for each algorithm to illustrate how the algorithm works. Each step in the algorithm is represented by a circled number in the figure. For example, Step 1 in the algorithm coincides with in the figure.

2.2.2. *Movement operation.* Next, we describe the movement operation of the MA controlling mechanism. Figures 3 and 4 are two cases that could occur, while the MU moves. Figure 3 shows the MU's movement inside the region of level $R_\ell^{b_{old}}$ (which is equal to $R_\ell^{b_{new}}$). In this case, the MA does not need to move. Figure 4 shows the MU moving out of $R_\ell^{b_{old}}$. The MA, therefore, has to be moved to a new location.

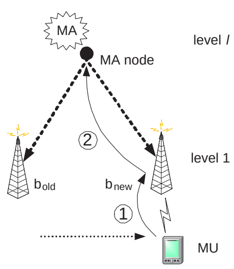


FIGURE 3. Scenario 1 of the MA's movement operation

Algorithm 2 shows the above two cases in detail. The circled numbers in the figure correspond to the steps of the algorithm.

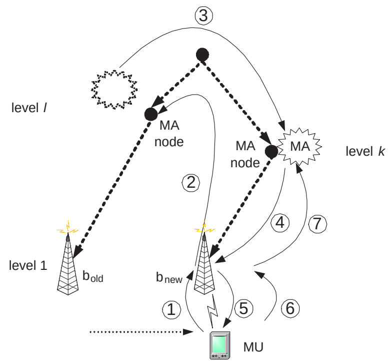


FIGURE 4. Scenario 2 of the MA's movement operation

Algorithm 2 Movement Operation

1: The MU moves to b_{new} from b_{old}, and sends the movement message to b_{new}.
2: b_{new} forwards this movement message to the MA. Assume that the MA is at level ℓ.
3: The MA updates the MU's location in MU_address.
If the MA node is the common ancestor of b_{new} and b_{old}, then do nothing; else the MA calls for an analysis and finds the best fitted location to stay (to be introduced in Section 3), and then moves to the new MA node which is assumed to be at level k.
4: The MA sends an update message to b_{new} through the wired network in order to inform the MU to update the location of MA in MA_address.
5: b_{new} forwards the update message to the MU through the wireless network.
6: The MU sends the acknowledgement to b_{new}.
7: b_{new} forwards the acknowledgement to the MA.

Note that the MA is always the ancestor of the MU in the network structure, no matter where the MU moves to. In this scenario, the new MA location needs to be determined. The new MA location at a different level of the network, incurs different costs for communicating with the MU and the server. Therefore, the best level for the MA should be carefully computed (Step 3 of the algorithm). We devised a model for this problem and an evaluation scheme for finding the optimal level for the MA, so that the total communication overhead is minimised. They will be presented in the next section, and the whole analysis in that section can be implemented as a function called by Step 3 of the algorithm.

3. **Analysis of the New Mechanism.** In this section, we analyse the proposed mechanism. We first introduce the parameters used in the model. Next, we describe the movement constraint of the MA. Finally, the cost functions for describing the proposed mechanism are derived.

3.1. **Parameters.** The parameters used in our cost models are listed in Figure 5. Fine-granularity parameters are used in describing the mobile computing systems.

In order to represent the MU's request and movement behavior, we introduce a parameter, $\mathcal{MIR}$ which is defined as the ratio of the number of movement operations to the number of request operations within a certain time period, i.e.,

$$\mathcal{MIR} = \frac{\text{frequency of movement operations}}{\text{frequency of request operations}} = \frac{\text{number of movement operations}}{\text{number of request operations}}$$

Parameter	Description
System parameters	
$\mathcal{MIR}$	The MU's mobility/request ratio.
$\mathcal{R}_x$	The size of x in the request operation, $x \in \{request, result\}$.
$\mathcal{M}_y$	The size of y in the movement operation, $y \in \{m, u, a\}$.
$\mathcal{A}$	The size of a MA.
α	The communication cost of transferring 1KB data through the wireless network.
β_ℓ	The communication cost of transferring 1KB data between the level ℓ and $\ell + 1$ of the wired network.
P_ℓ	The probability that a MU moves inside the boundary of $R_\ell^{b_{MU}}$.
Q_ℓ	The probability that a MU moves out of the boundary of $R_\ell^{b_{MU}}$.
μ_ℓ	The probability that a MU stays within $R_\ell^{b_{MU}}$ when a MU moves out of $R_{\ell-1}^{b_{MU}}$.
Performance Metrics and Strategy-related parameters	
L	The farthest level that an MA can arrive.
C_R^ℓ	The request cost that a MA sends a request at the level ℓ.
C_m^ℓ	The movement cost that a MA moves inside the region of level ℓ.
$C_M^{(\ell,h)}$	The movement cost a MA moves out the region of level ℓ throuth level h.
C_{total}^ℓ	The total communication cost of a MA at the level ℓ.

FIGURE 5. Parameters and their meanings

MUs with large $\mathcal{MIR}$ means they infrequently use their mobile service while they move. On the contrary, MUs with small $\mathcal{MIR}$ means they are workaholic type of users.

In request operations, two kinds of data are considered. They include the size of MU's request (denoted as $\mathcal{R}_{request}$) which is used in Step 1 and Step 2 of the request algorithm, and the size of result (denoted as $\mathcal{R}_{result}$) which is used in Step 4 and Step 5. In most cases, $\mathcal{R}_{request}$ is smaller than $\mathcal{R}_{result}$. In movement operations, we consider the MA size ($\mathcal{A}$) and three kinds of messages: (i) messages that the MU sends to the MA ($\mathcal{M}_m$), which is used in Step 1 and Step 2 of the movement algorithm; (ii) messages that the MA sends to the MU ($\mathcal{M}_u$), which is used in Step 4 and Step 5; and (iii) acknowledgement messages that the MU sends to the MA ($\mathcal{M}_a$), which is used in Step 6 and Step 7. The size of these three kinds of messages is usually small, about dozens of kilo-bytes.

In this system, the communication cost of transferring 1KB size of data in wireless network is denoted as α; and the communication cost of transferring 1KB size of data between level ℓ and $\ell + 1$ is denoted as β_i.

We also use three parameters (including P_ℓ, Q_ℓ and μ_ℓ) to describe how the MU moves across cells. However, explanations of these parameters will involve the concept of computing total communication cost. Therefore, these three parameters will be examined in detail, in Section 3.3.

3.2. **Movement constraint of MA.** In principle, a MA can stay at any MA node. However, staying at a MA node which is too far away from the MU is unpractical, because it would incur large communication cost. In this subsection, we derive the farthest level where a MA can reside.

Wireless communication normally incurred high cost in the past and, unfortunately, nothing much could be done to reduce the cost. The cost of wired communication, on the other hand, can be reduced by moving the MA to a proper level at the network hierarchy. Therefore, the best we can do is to keep the cost of wired communication lower than that of the wireless part. In other words, for data of size $\mathcal{D}$ to be transferred, we have the following constraint:

$$\begin{aligned} \mathcal{D} \cdot \alpha &\geq \sum_{i=1}^{\ell-1} \mathcal{D} \cdot \beta_i \\ \Longrightarrow \alpha &\geq \sum_{i=1}^{\ell-1} \beta_i \end{aligned}$$

where ℓ is the level a MA moves to. It is interesting to note that this inequation has nothing to do with the size of data ($\mathcal{D}$). And the relationship between α and β_i is dependent solely on ℓ. Assume that L is the highest allowable level that the MU can moves to, so that the above constraint is still maintained. Then, we can further infer that the following inequation should be true.

$$\sum_{i=1}^{L} \beta_i > \alpha \geq \sum_{i=1}^{L-1} \beta_i$$

From this inequation, L can be determined.

Considering a case that the communication costs of transferring a data unit at each level are the same, i.e., $\beta_1 = \beta_2 = \cdots = \beta_\ell = \beta$. Then, the in equation can be rewritten as:

$$\begin{aligned} &\sum_{i=1}^{L} \beta_i > \alpha \geq \sum_{i=1}^{L-1} \beta_i \\ \Longrightarrow &\sum_{i=1}^{L} \beta > \alpha \geq \sum_{i=1}^{L-1} \beta \\ \Longrightarrow &\, L \cdot \beta > \alpha \geq (L-1) \cdot \beta \end{aligned}$$

After simple algebra calculations, we have:

$$L > \frac{\alpha}{\beta} \text{ and } L \leq \frac{\alpha}{\beta} + 1$$

Therefore, L can be represented as follows:

$$L = \left\lfloor \frac{\alpha}{\beta} \right\rfloor + 1$$

This equation indicates that: the larger the difference between the communication costs of wired and wireless networks, the farther the MA can be from the MU, as we have known. However, most importantly, it quantifies their relationship and shows the exact number of levels that the MA can profitably reach.

3.3. **Derivation of cost functions.** In this subsection, we derive the cost equations for the proposed mechanism. When the MA stays at a different level, it would incur a different cost for request operations and movement operations and, therefore, a different total cost. That is, the total cost is a function of the MA's resident level ℓ. From the viewpoint of the request operations, it is beneficial if the MA is close to the MU, but from the view point of the movement operations, it is beneficial if the MA is far away from the MU. Therefore, a tradeoff exists in this mechanism. Our goal is to find the optimal ℓ for a MA.

As there are only two types of operations: request and movement operations, the total cost can be represented on the per-request basis as follows.

$$C^{\ell}_{total} = \text{(the cost of a request operation)} + \mathcal{MIR} \cdot \text{(the cost of a movement operation)}$$

The cost of a request operation that a MA is at level ℓ is denoted as C^{ℓ}_{R}. In terms of a movement operation, it can be divided into two subcases:

1) the movement of the MU is still within $R^{b_{MU}}_{\ell}$: the communication cost of the movement operation is denoted as C^{ℓ}_{m}.
2) the MU moves out of $R^{b_{MU}}_{\ell}$: the level of the least common ancestor of the newly arrived cell and the old cell affects the communication cost. We assume the least common ancestor of MU's new cell and old cell is at level h, and denote such movement operation cost as $C^{(\ell,h)}_{M}$.

We employ two probabilities, P_ℓ and Q_ℓ, to model the presence of the above two cases. P_ℓ stands for the probability of subcase 1 and Q_ℓ stands for the probability of subcase 2. We can then infer that $P_1 = 0$ and $Q_1 = 1$, because a movement of the MU inherently implies that the MU has left the current MSS region and entered into the region of another MSS. If the MU has physically moved, but is still within the region of the same MSS, then to the system the MU is unmoved at all. Q_ℓ can be easily expressed as follows.

$$Q_\ell = \begin{cases} 1, & \text{if } \ell = 1. \\ 1 - P_\ell, & \text{if } \ell \geq 2. \end{cases}$$

P_ℓ can be obtained by adding the probability that the MU's movement is within $R^{b_{MU}}_{\ell-1}$ and the probability that the MU moves out of $R^{b_{MU}}_{\ell-1}$ but still inside $R^{b_{MU}}_{\ell} - R^{b_{MU}}_{\ell-1}$. Figure 6 illustrates the relationship between $R^{b_{MU}}_{\ell}$ and $R^{b_{MU}}_{\ell-1}$. The probability that the MU moves inside $R^{b_{MU}}_{\ell-1}$ is $P_{\ell-1}$. In the situation that the MU moves out of $R^{b_{MU}}_{\ell-1}$, we assume μ_ℓ is the probability that the MU would stay inside $R^{b_{MU}}_{\ell}$. Therefore, $(Q_\ell \cdot \mu_\ell)$ is the probability that the MU stays inside the region $R^{b_{MU}}_{\ell+1} - R^{b_{MU}}_{\ell}$.

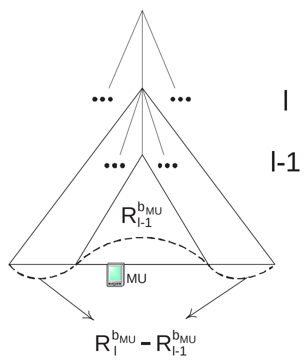


FIGURE 6. An example of $R^{b_{MU}}_{\ell} - R^{b_{MU}}_{\ell-1}$

Figure 7 shows the probabilities that the MU moves out of $R^{b_{MU}}_{\ell-1}$. Therefore, P_ℓ can be represented as follows.

$$P_\ell = \begin{cases} 0, & \text{if } \ell = 1, \\ P_{\ell-1} + Q_{\ell-1} \cdot \mu_i = \sum_{i=2}^{\ell} Q_{i-1} \cdot \mu_i, & \text{if } \ell \geq 2, \end{cases}$$

where $\sum_{i=2}^{\ell} Q_{i-1} \cdot \mu_i$ is the result after $P_{\ell-1}$ is replaced by $P_{\ell-2} + Q_{\ell-2} \cdot \mu_{\ell-1}$, and recursively until all P_i is replaced, $1 \le i \le \ell$.

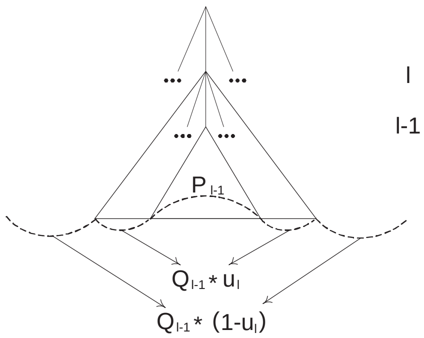


FIGURE 7. Probabilities that the MU moves out of $R_{\ell-1}^{b_{MU}}$

Utilizing the above probability model, the total cost can be rewritten as follows.

$$\begin{aligned} C_{total}^{\ell} &= \text{(the cost of a request operation)} + \mathcal{MIR} \cdot \text{(the cost of a movement operation)} \\ &= C_R^{\ell} + \mathcal{MIR} \cdot \left(P_\ell \cdot C_m^{\ell}\right) + \mathcal{MIR} \cdot \left(\sum_{h=\ell+1}^{L} (Q_{h-1} \cdot \mu_h) \cdot C_M^{(\ell,h)} \right) \end{aligned}$$

C_R^{ℓ} in this total cost can be calculated in this way. In the request operation in Algorithm 1, there are five steps related to C_R^{ℓ}. Step 1 and Step 2 deliver the MU's request to the MA, therefore, they spend the communication cost $\mathcal{R}_{request} \cdot \alpha$ and $\sum_{i=1}^{\ell-1} \mathcal{R}_{request} \cdot \beta_i$, respectively. Step 3 processes the receiving request on the MA, thus it costs nothing in terms of communication. Step 4 and Step 5 transfer the request result from the MA to the MU, therefore, they spend the communication cost $\sum_{i=1}^{\ell-1} \mathcal{R}_{result} \cdot \beta_i$ and $\mathcal{R}_{result} \cdot \alpha$, respectively. We obtain the request operation cost as follows:

$$C_R^{\ell} = \mathcal{R}_{request} \cdot \alpha + \left(\sum_{i=1}^{\ell-1} \mathcal{R}_{request} \cdot \beta_i \right) + \left(\sum_{i=1}^{\ell-1} \mathcal{R}_{result} \cdot \beta_i \right) + \mathcal{R}_{result} \cdot \alpha \tag{1}$$

In the case that the MU's move is inside $R_{\ell}^{b_{MU}}$, only a movement message is delivered from the MU to the MA. Thus, C_m^{ℓ} is as follows:

$$C_m^{\ell} = \mathcal{M}_m \cdot \alpha + \left(\sum_{i=1}^{\ell-1} \mathcal{M}_m \cdot \beta_i \right) \tag{2}$$

The movement operation that the MU moves out of $R_{\ell}^{b_{MU}}$ is given in Algorithm 2. Step 1 and Step 2 are to send the movement messages from the MU to the MA. The communication cost of Step 1 is $\mathcal{M}_m \cdot \alpha$ because the communication is through the wireless network. In Step 2, a message of size $\mathcal{M}_m$ is sent to the MA through the least common ancestor (at level h) of the MU and the MA because the MU moves out of $R_{\ell}^{b_{MU}}$. The communication cost of Step 2 is $\sum_{i=1}^{h-1} \mathcal{M}_m \cdot \beta_i + \sum_{i=\ell}^{h-1} \mathcal{M}_m \cdot \beta_i$. Step 3 is to move the MA to the new MA node. This step costs $\sum_{i=\ell}^{h-1} 2 \cdot \mathcal{A} \cdot \beta_i$. Step 4 and Step 5 are to deliver a message from the MA to the MU. The costs of these two steps are $\sum_{i=1}^{\ell-1} \mathcal{M}_u \cdot \beta_i$ and $\mathcal{M}_u \cdot \alpha$, respectively. Step 6 and Step 7 are to send an acknowledgement from the MU to the MA. They cost $\mathcal{M}_a \cdot \alpha$ and $\sum_{i=1}^{\ell-1} \mathcal{M}_a \cdot \beta_i$, respectively. Summing up the costs of all steps is $C_M^{(\ell,h)}$, we have

$$C_M^{(\ell,h)} = \mathcal{M}_m \cdot \alpha + \left(\sum_{i=1}^{h-1} \mathcal{M}_m \cdot \beta_i\right) + \left(\sum_{i=\ell}^{h-1} \mathcal{M}_m \cdot \beta_i\right) + \left(\sum_{i=\ell}^{h-1} 2 \cdot \mathcal{A} \cdot \beta_i\right) + \left(\sum_{i=1}^{\ell-1} \mathcal{M}_u \cdot \beta_i\right) + \mathcal{M}_u \cdot \alpha + \mathcal{M}_a \cdot \alpha + \left(\sum_{i=1}^{\ell-1} \mathcal{M}_a \cdot \beta_i\right) \quad (3)$$

The total cost is, therefore,

$$C_{total}^{\ell} = \text{Equation (1)} + \mathcal{MIR} \cdot \Big(P_\ell \cdot \text{Equation (2)}\Big) + \mathcal{MIR} \cdot \left(\sum_{h=\ell+1}^{L} (Q_{h-1} \cdot \mu_h) \cdot \text{Equation (3)}\right)$$

In most cases, the sizes of $\mathcal{M}_u$, $\mathcal{M}_m$, $\mathcal{M}_a$ and $\mathcal{R}_{request}$ are much smaller than the sizes of $\mathcal{A}$ and $\mathcal{R}_{result}$. Therefore, $\mathcal{A}$ and $\mathcal{R}_{result}$ are the dominators of the total cost among the six parameters. It can be seen that C_{total}^{ℓ} is a complex function of ℓ. To find a closed expression of the optimal ℓ is very hard. In this paper, we use simulation to look for the optimal ℓ. The results are presented in the next section.

4. Performance Evaluation.

4.1. Performance metrics and experimental setup.

In the performance study, we try to answer the following questions about the performance of the proposed MA controlling mechanism:

- How does the MU's moving behavior ($\mathcal{MIR}$) affect the resident level of the MA?
- Does the probability μ_i affect the resident level of the MA?
- What is the effect of the sizes of the MA and the request result ($\mathcal{A}$ and $\mathcal{R}_{result}$)?
- What is the effect of the communication costs of wired and wireless networks (α and β)?

To simplify our task without the loss of generality, we treat the communication at all levels of the wired network as of equal bandwidth, i.e., $\beta_1 = \beta_2 = \cdots = \beta_L = \beta$. In addition, the probabilities that the MU could stay inside Z_i at all levels are the same, i.e., $\mu_1 = \mu_2 = \cdots = \mu_L = \mu$. Our parameter settings are chosen as close to the real situation as possible. The default values of the parameters used in our evaluation are given in Figure 8.

Parameter	*Default value*
μ	0.5
$\mathcal{A}$	10000 KB
α	10
β	1
$\mathcal{R}_{result}$	2000 KB
$\mathcal{R}_{request}$	10 KB
$\mathcal{MIR}$	1
$\mathcal{M}_m$	10 KB
$\mathcal{M}_u$	10 KB
$\mathcal{M}_a$	10 KB

FIGURE 8. Default values of parameters

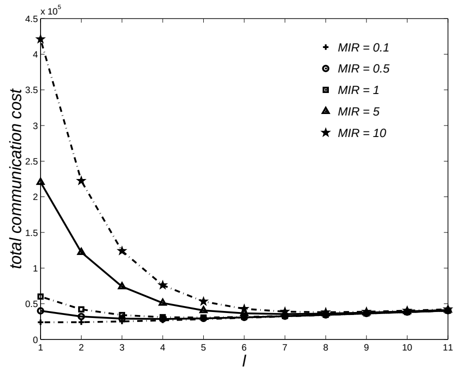


FIGURE 9. The effect of $\mathcal{MIR}$

4.2. **Experiment 1: effect of $\mathcal{MIR}$.** Figure 9 shows the effect of $\mathcal{MIR}$ to total cost. The horizontal-axis in the figure is the MAs' resident level, which varies from 1 to 11. The vertical-axis in the figure is the total communication cost. From these curves in Figure 9, we observe that the greater the $\mathcal{MIR}$, the greater the optimal ℓ. For clarity, we further plot the optimal level for different $\mathcal{MIR}$'s in Figure 10. The lowest points of $\mathcal{MIR} = \{0.1, 0.5, 1, 5, 10\}$ curves show the optimal ℓ, and they are $\{1, 4, 5, 7, 8\}$, respectively. This is because when $\mathcal{MIR}$ is large, the number of a MU's movement operations is more than the number of the MU's request operations. Therefore, it is beneficial for the MA to stay at a higher level. Similarly, for those MUs whose $\mathcal{MIR}$ is small, it is beneficial for their MA to stay at a lower level. Another observation is that $\mathcal{MIR}$ is quite sensitive to ℓ when $\mathcal{MIR}$ is large. This is because the size of a MA is larger than the size of the request result. Therefore, if the MA stays close to its MU when $\mathcal{MIR}$ is large, high communication cost could be incurred, because of the frequent movement operations.

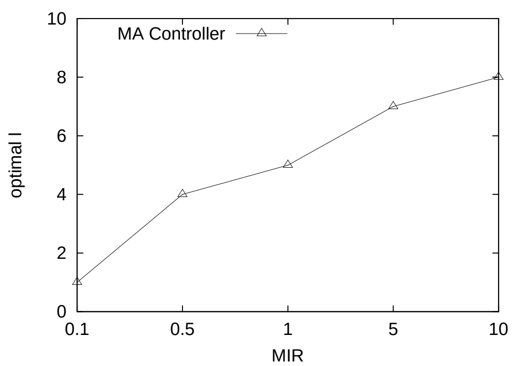


FIGURE 10. The optimal level for different $\mathcal{MIR}$'s

4.3. **Experiment 2: effect of μ.** Figure 11 shows the effect of μ to the total communication cost. From the results, we found from this experiment that the smaller the μ, the greater the optimal ℓ. Similar to the first experiment, we further plot the optimal level for different $\mathcal{MIR}$'s in Figure 12. The optimal ℓ's for the curves of $\mu = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ occur at $\{11, 9, 5, 3, 2\}$, respectively. This is because a small μ indicates that the MU will easily move out Z_i. Therefore, the MA staying at the higher level would be better than one staying at a lower level. Also, all curves almost merge when $\ell = 11$. This is mainly because the probability that a MU moves out of the region of a MA node at as high a level as 11 would be very small. Therefore, the movement operation costs of all curves become almost the same.

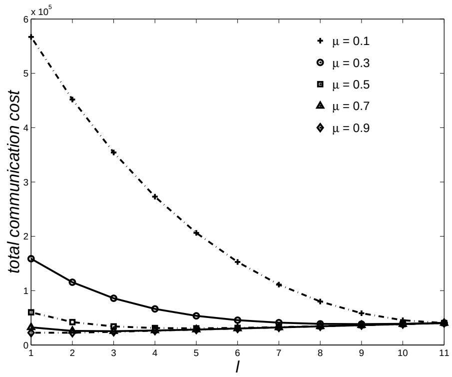


FIGURE 11. The effect of μ

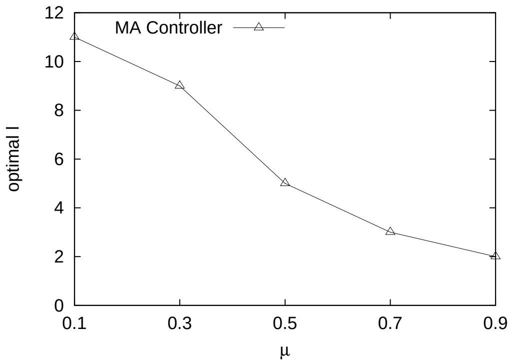


FIGURE 12. The optimal level for different μ's

Notice that the above two experiment results show a vigorous motivation of the practical use of the theoretic results. From Figures 10 and 12, we can see the curves of optimal ℓ to $\mathcal{MIR}$ and μ are not regular functions. That is, the optimal level does not directly proportional or inverse-proportional to the given environment parameters. Hence, for those mobile agent-based applications, our derived theoretic result can give beneficial assistance for the mobility component of mobile agents to obtain the optimal levels for MAs, and thus, a great amount of communication cost can be saved.

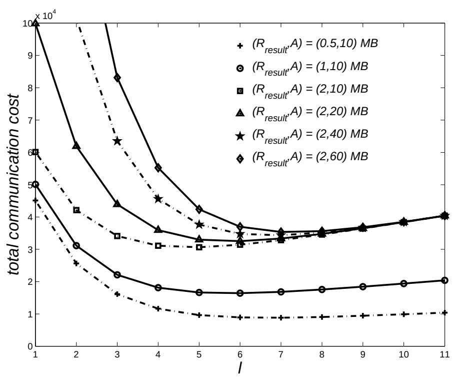
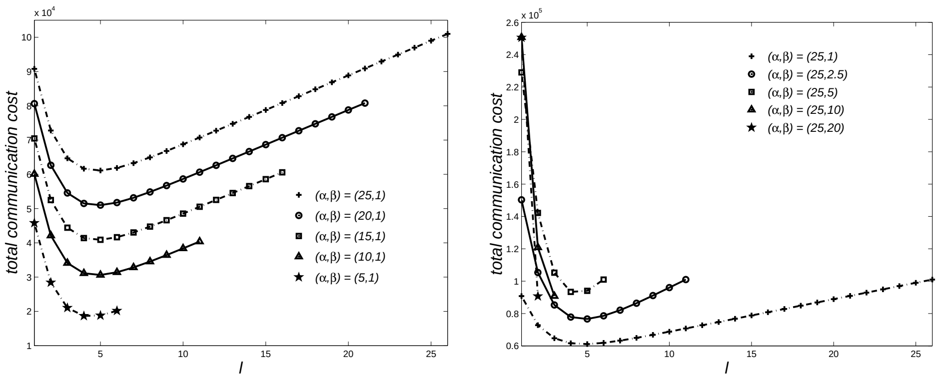


FIGURE 13. The effect of $\mathcal{R}$ and $\mathcal{A}$.

4.4. **Experiment 3: effect of $\mathcal{R}_{result}$ and $\mathcal{A}$.** In the third experiment, we study the effects of $\mathcal{R}_{result}$ and $\mathcal{A}$ to total cost, and the result is shown in Figure 13. The curves in the figure can be roughly divided into two groups: one with the curves of $(\mathcal{R}_{result}, \mathcal{A}) = \{(0.5, 10), (1, 10), (2, 10)\}$ in which only $\mathcal{R}_{result}$ varies; the other with the curves of $(\mathcal{R}_{result}, \mathcal{A}) = \{(2, 10), (2, 20), (2, 40), (2, 60)\}$ in which only $\mathcal{A}$ varies. The optimal ℓ's of $(\mathcal{R}_{result}, \mathcal{A}) = \{(0.5, 10), (1, 10), (2, 10)\}$ appear at $\{7, 6, 5\}$, respectively. The greater the $\mathcal{R}_{result}$, the smaller the optimal ℓ. This is because when the result size is large, a MA at a lower ℓ sends the data with a smaller cost. The optimal ℓ's of the second group $(\mathcal{R}_{result}, \mathcal{A}) = \{(2, 10), (2, 20), (2, 40), (2, 60)\}$ occur at $\{5, 6, 7, 7\}$, respectively. The greater the value of $\mathcal{A}$, the larger the optimal ℓ. This is because when a large sized MA resides at a higher ℓ, the number of movement operations could be reduced.



(a) The effect of α (b) The effect of β

FIGURE 14. The effect of communication costs

4.5. **Experiment 4: effect of α and β.** In the last experiment, we study the effect of α and β to total cost, and the result is shown in Figures 14(a) and 14(b), respectively. The curves in Figure 14 have different length. This is because the ratio of the communication costs of wired and wireless networks determines the maximal possible MA's resident level. As we discussed in Section 3.2, the maximal MA's resident level is $\lfloor \alpha/\beta \rfloor + 1$. They are drawn according to this rule.

In Figure 14, we find the optimal ℓ occurs at about 5, and is not affected by different α and β values. Increasing α or β only increases total communication cost. We also note that if β approaches α (e.g., $(25, 10)$ and $(25, 20)$ in Figure 14(b)), increasing one level could significantly reduce the total cost. This is because the movement operation cost is greatly reduced.

5. **Savings under Optimal ℓ and Further Findings.** We have studied the effect of different factors separately on the proposed strategy, and found that the communication cost is very sensitive to these factors. In the following, we go one step further, to study the percentage of communication cost that can be saved when a good location for the MA is chosen. The cost saving is studied based on two criteria.

1) **Maximum cost-saving ratio** (denoted as $\mathcal{S}_{\max}$): is defined as

$$\mathcal{S}_{\max} = \frac{\max(C^1_{total}, \cdots, C^L_{total}) - C^{\ell_{opt}}_{total}}{C^{\ell_{opt}}_{total}}$$

where ℓ_{opt} is the optimal ℓ.

2) **Average cost-saving ratio** (denoted as $\mathcal{S}_{avg}$): is defined as

$$\mathcal{S}_{avg} = \frac{1}{L} \cdot \left(\sum_{i=1}^{L} \frac{C_{total}^{i} - C_{total}^{\ell_{opt}}}{C_{total}^{\ell_{opt}}} \right)$$

By using these two criteria, we compute for each curve of each experiment the cost-saving ratios and show them in Figure 15. Figures 15(a)-15(e) respectively show the results by varying one parameter and fix the others. The default parameter values are $\mathcal{MIR} = 1$, $\mu = 0.5$, $(\mathcal{R}_{result}, \mathcal{A}) = (2, 10)$ and $(\alpha, \beta) = (10, 1)$. They are also marked by a superscript '$*$' in the tables. These default values are at about the middle of their corresponding varying ranges in the experiments so that the results will not prejudice our case. A quick look at the results tells us that both the $\mathcal{S}_{\max}$ and $\mathcal{S}_{avg}$ are quite large percentages for all cases. Almost all of them are over 20% and many of them even higher than 50%. At the bottom of each table, an average of the cost-saving ratios listed above it in that table is given. Even the average cost reductions of all these scenarios are from as high as tens to a few hundred percent. This means that there is significant saving, if the location of the MA is carefully selected.

	cost-saving ratio	
$\mathcal{MIR}$	$\mathcal{S}_{\max}$	$\mathcal{S}_{avg}$
0.1	67.51%	28.58%
0.5	40.78%	18.10%
1*	96.15%	22.28%
5	520.77%	88.00%
10	1006.80%	171.89%
average	346.40%	65.77%

(a) Cost-saving ratio of $\mathcal{MIR}$

	cost-saving ratio	
μ	$\mathcal{S}_{\max}$	$\mathcal{S}_{avg}$
0.1	1306.04%	427.52%
0.3	314.34%	71.51%
0.5*	96.15%	22.28%
0.7	59.40%	25.81%
0.9	79.88%	35.95%
average	371.16%	116.61%

(b) Cost-saving ratio of μ

	cost-saving ratio	
$(\mathcal{R}_{result}, \mathcal{A})$	$\mathcal{S}_{\max}$	$\mathcal{S}_{avg}$
$(0.5, 10)$	410.65%	69.48%
$(1, 10)$	205.08%	36.82%
$(2, 10)^*$	96.15%	22.28%
$(2, 20)$	206.95%	37.08%
$(2, 40)$	422.18%	71.34%
$(2, 60)$	632.03%	106.75%
average	328.84%	57.29%

(c) Cost-saving ratio of $(\mathcal{R}_{result}, \mathcal{A})$

	cost-saving ratio	
(α, β)	$\mathcal{S}_{\max}$	$\mathcal{S}_{avg}$
$(25, 1)$	65.25%	29.03%
$(20, 1)$	58.39%	26.33%
$(15, 1)$	72.33%	23.43%
$(10, 1)^*$	96.15%	22.28%
$(5, 1)$	145.28%	36.53%
average	87.48%	27.52%

(d) Cost-saving ratio of (α, β): fixed β

	cost-saving ratio	
(α, β)	$\mathcal{S}_{\max}$	$\mathcal{S}_{avg}$
$(25, 1)$	65.25%	29.03%
$(25, 2.5)$	96.15%	22.28%
$(25, 5)$	145.28%	36.53%
$(25, 10)$	176.21%	69.77%
$(25, 20)$	176.34%	88.17%
average	131.85%	49.15%

(e) Cost-saving ratio of (α, β): fixed α

FIGURE 15. Maximum cost saving ratio and average cost saving ratio of our experiments

From Figure 15(a), we observe that for a large $\mathcal{MIR}$ ($\mathcal{MIR} \geq 5$), i.e., a mobile user of high mobility, a good selection of MA's location greatly reduces the communication cost. Therefore, mobile users on a highway can be greatly benefited by having such a mechanism.

Choosing MA's resident node is also necessary in a metropolitan area. The number of base stations in a metropolitan area is normally more than that in a suburban area so as to serve crowded MUs. Therefore, the cell region is normally small in a metropolitan area. It implies that the probability of a MU to cross cells becomes high, (i.e., μ is small). Figure 15(b) shows that our proposed mechanism performs particularly well when μ is small.

The amount of wasted communication cost resulting from movement/request operations depends not only on $\mathcal{MIR}$ and μ, but also on the size of the MA ($\mathcal{A}$) and the size of the request result ($\mathcal{R}_{result}$). Figure 15(c) shows that when a movement/request operation occurs, on average 57.29% communication cost is saved and 328.84% could be saved at the most. Therefore, if a good MA resident node is chosen, at least half of communication cost can be easily reduced.

Figures 15(d) and 15(e) show the bandwidth effect of the wired network and the wireless network. Our results indicate that a careful selection of MA's location can dramatically reduce communication cost for various types of networks, particularly, for the slow (wireless) networks (referring to the $(25, 20)$ tuple in Figure 15(e)). Therefore, the lower (including less reliable) the network bandwidth, the more important the location of the MA.

6. **Implementation Issues: Considering Duplicate Data Items in a MA Node.** In a MA node, some data items carried by different MAs could be the same. In this section, three methods of managing such duplicate data items are designed.

In order to formally compare these methods, some parameters are introduced first. Assume that n MAs are located in a node, and MA_i has u_i data items, where $1 \leq i \leq n$. The storage cost for MA's execution codes is about the same, and is denoted as γ_{MA}. There are totally q distinct data items, $p_1, p_2, \ldots, p_q$, carried by n MAs in a node. For simplicity in comparison, we assume these data items are of the same size, denoted as γ_d. We also assume o_i is the number of MAs sharing data item p_i, where $1 \leq i \leq q$.

The first method of managing MAs' data items, the MA-majored method (MAM), is to store the MA and *data_items* in a table, and assign MA as the key attribute of the table. Figure 16 is an example of such a table. Two MAs (MA_1 and MA_2) are in the table, and they own data items p_1, p_2, p_3 and p_1, p_2, p_4, respectively. This method is intuitive, and ignores the problem of data duplication. Thus duplicate data items are repeatedly stored in the table. The total storage cost of MAM (denoted as Φ_{MAM}) is the sum of sizes of all MAs and their data items, i.e.,

$$\Phi_{MAM} = n \cdot \gamma_{MA} + \left(\sum_{i=1}^{n} u_i \right) \cdot \gamma_d$$

$\underline{MA}$	*data_items*
MA_1	p_1, p_2, p_3
MA_2	p_1, p_2, p_4

FIGURE 16. A storage scenario of MAM

The second method, the data-item-majored method (DIM), is similar to MAM but with a different key attribute. The key attribute of the table of DIM is *data_items*, rather than MA. Figure 17 is a storage scenario of DIM, and the MAs and their data items are the same as in the above example. In this method, the data items will not be stored duplicately, as in MAM. Thus, the total storage cost of DIM (denoted as Φ_{DIM}) is the sum of all distinct data item sizes and all MA sizes (including duplicate MAs), i.e.,

$$\Phi_{DIM} = q \cdot \gamma_d + \left(\sum_{i=1}^{q} o_i \right) \cdot \gamma_{MA}$$

Notice that the total number of duplicate MAs can be counted as the sum of all o_i.

data_item	MA
p_1	MA_1, MA_2
p_2	MA_1, MA_2
p_3	MA_1
p_4	MA_2

FIGURE 17. A storage scenario of DIM

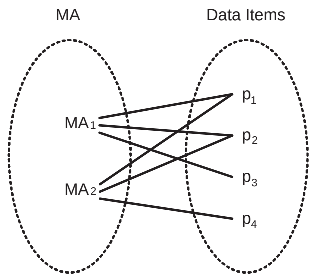


FIGURE 18. A storage scenario of GM

The third method of managing MAs and their data items is a graph-based method (GM), and an example is shown in Figure 18. Its basic idea of avoiding duplication is that all MAs and their relationships with data items are managed by a bipartite graph in a MA node. One set in this graph has all the MA codes, and the other has all the data items. An edge existing between MA_i and data item p_j if p_j is one of the data items used in MA_i. Each edge is represented by two identifiers: a MA ID and a data item ID. Thus, its size is very small and becomes negligible. In the bipartite graph, the degree of MA_i represents the number of data items used in MA_i, and the degree of p_j represents the number of MAs sharing p_j. The total storage cost of GM (denoted as Φ_{GM}) is the sum of the sizes of all distinct data items and the sizes of distinct MAs' codes. That is,

$$\Phi_{GM} = q \cdot \gamma_d + n \cdot \gamma_{MA}$$

Since the storage cost functions of three methods are derived, we then compare these methods from the storage size aspect. Among these three methods, we find that GM uses the least amount of storage. This is because only distinct MAs and distinct *data_item*s are used in GM, but both MAM and DIM need to store additional duplicates. The difference of the storage size between MAM and GM is

$$\Phi_{MAM} - \Phi_{GM} = \left(\sum_{i=1}^{n} u_i - q \right) \cdot \gamma_d$$

and the difference of the storage size between MAM and GM is

$$\Phi_{DIM} - \Phi_{GM} = \left(\sum_{i=1}^{q} o_i - n \right) \cdot \gamma_{MA}$$

Incidentally, we make a simple comparison on the search speed of these methods. One may think that the savings on storage cost may be traded with the search speed as our intuition tells. Our study, however, indicates that GM's search speed is also the best. Two search functions ("search a MA" and "search a data item") are considered here because these two functions are the most frequently used in the MA controlling mechanism and in answering the requests of MUs. Figure 19 lists the comparisons of search complexity on the three methods. MAM is beneficial for searching a MA because MA is the key attribute in this method (that is, no duplicate on this column). Thus, MAM only spends $\mathcal{O}(n)$ time on searching a MA (without the support of index). On the other hand, MAM needs to spend $\mathcal{O}(\sum_{i=1}^{n} o_i)$ computing cost on searching a data item because of the duplicates. Comparing with MAM, DIM has a better ability on searching a data item, but behind MAM on searching a MA. The computation complexity of searching a data item of DIM is $\mathcal{O}(q)$ because the data item is the key attribute. When DIM searches a MA, however, all MAs (including duplicates) need to be checked. Thus the complexity of searching a MA of DIM is $\mathcal{O}(\sum_{i=1}^{q} o_i)$. GM records only distinct MAs and distinct data items, hence, it is good at both search functions. The complexities of searching a MA and searching a data items are $\mathcal{O}(n)$ and $\mathcal{O}(q)$, respectively. In summary, comparing the three methods on storage size and searching ability, GM is the most suitable structure to manage MAs in a MA node for high duplication degree of data items.

	searching a MA	*searching a data item*
MAM	$\mathcal{O}(n)$	$\mathcal{O}(\sum_{i=1}^{n} u_i)$
DIM	$\mathcal{O}(\sum_{i=1}^{q} o_i)$	$\mathcal{O}(q)$
GM	$\mathcal{O}(n)$	$\mathcal{O}(q)$

FIGURE 19. A comparison of computation complexity of three methods

7. **Conclusions and Future Work.**

Wireless communication is an important technique in mobile client/server environments. However, retrieving remote data over the wireless network is still unreliable and bandwidth-limited. In this paper, in order to overcome the drawback of wireless communication, a MA controlling mechanism is designed for executing the communication-bound tasks of a MU. In this way, the use of wireless communication can be minimised. Unlike previous works, the MA here can stay at the most beneficial site, rather than always at level 1. The cost equations of the MA are derived using the analysis model. Our experimental results reveal that the MA controlling mechanism can save about 27% $\sim$ 116% communication cost. We are conducting further studies on performance evaluation and the MA movement in different network topologies.

Our future work would focus on the following directions. The first direction is to extend the optimal MA location issue to other new mobile computing environments, such as the wireless sensor networks. As we know, the wireless sensor network is one of most interesting research area in the last decade, and the data transmission in the sensor network is based on the tree structure in most cases [20, 21, 25]. In addition, mobile agent technologies are also adopted in the wireless sensor networks. Therefore, our study issue exists in the wireless sensor environments. The second direction is to consider the more complex behavior for the object movement and service activities. The more complex behavior can make the simulation results more close to the real-world scenarios. However, it also increase the difficulty of simulation techniques. In this work, we use mathematics techniques to derive the cost models for the proposed method. We are planing to adopt advanced mathematics tools, such as Stochastic Petri Net, to consider the complex behavior of the object movement and service activities.

Acknowledgement. Authors thank anonymous reviewers for their valuable comments on improving the paper quality. This work was partially supported by the National Science Council of Taiwan under grants NSC99-2221-E-218-035 and NSC100-2221-E-218-053-MY2.

REFERENCES

[1] V. Anantharam, M. L. Honig, U. Madhow and V. K. Kei, Optimization of a database hierarchy for mobility tracking in a personal communications network, *Performance Evaluation*, vol.20, pp.287-300, 1994.

[2] D. Barbara, Mobile computing and databases – A survey, *IEEE Transactions on Knowledge and Data Engineering*, vol.11, no.1, pp.108-117, 1999.

[3] B. R. Badrinath, A. Bakre, T. Imielinski and R. Marantz, Handling mobile clients: A case for indirect interaction, *Proc. of the 4th Workshop on Workstation Operation Systems*, Aigen, Austria, 1993.

[4] J. Bajo, J. M. Corchado, C. Pinzón, Y. de Paz and B. Pérez-Lancho, SCMAS: A distributed hierarchical multi-agent architecture for blocking attacks to databases, *International Journal of Innovative Computing, Information and Control*, vol.6, no.9, pp.3787-3817, 2010.

[5] B. Behdani, Z. Lukszo, A. Adhitya and R. Srinivasan, Agent-based modeling to support operations management in a multi-plant enterprise, *International Journal of Innovative Computing, Information and Control*, vol.6, no.7, pp.2873-2884, 2010.

[6] I.-R. Chen, T.-M. Chen and C. Lee, Agent-based forwarding strategies for reducing location management cost in mobile networks, *ACM/Baltzer Journal on Mobile Network and Application*, vol.6, no.2, pp.105-115, 2001.

[7] C.-C. Chen and C. Lee, Location management strategies for reducing both movement cost and locating cost simultaneously, *Proc. of the 1st International Conference on Mobile Data Access*, Hong Kong, pp.209-219, 1999.

[8] P. S. Chen, S.-H. Li and Y.-K. Liu, Scheduling the access to multi-level secure databases in a wireless network environment, *International Journal of Innovative Computing, Information and Control*, vol.6, no.12, pp.5381-5403, 2010.

[9] A. Fox and E. A. Brewer, Reducing WWW latency and bandwidth requirements by real-time distillation, *Proc. of the 5th International World Wide Web Conference*, Paris, France, 1996.

[10] R. S. Gray, G. Cybenko, D. Kotz and D. Rus, Mobile agents: Motivations and state of the art, *Handbook of Agent Technology*, 2002.

[11] R. S. Gray, G. Cybenko, D. Kotz, R. A. Peterson and D. Rus, D'Agents: Applications and performance of a mobile-agent system, *Software – Practice and Experience*, vol.32, no.6, pp.543-573, 2002.

[12] R. S. Gray, D. Kotz, R. A. Peterson et al., Mobile-agent versus client/server performance: Scalability in an information-retrieval task, *Proc. of the 5th IEEE International Conference on Mobile Agents*, Atlanta, GA, USA, pp.229-243, 2001.

[13] C.-H. Liu, Y.-F. Chung, T.-S. Chen and S.-D. Wang, An ID-based access control in a hierarchical key management for mobile agent, *International Journal of Innovative Computing, Information and Control*, vol.7, no.3, pp.1443-1456, 2011.

[14] N. W. Lo and K.-H. Yeh, A novel authentication scheme for mobile commerce transactions, *International Journal of Innovative Computing, Information and Control*, vol.6, no.7, pp.3093-3103, 2010.

[15] T. Imielinski and B. Badrinath, Mobile wireless computing: Challenges in data management, *Communications of the ACM*, vol.37, no.10, 1994.

[16] M. J. Franklin, Challenges in ubiquitous data management, *Proc. of Informatics: 10 Years Back, 10 Years Ahead*, London, UK, 2001.

[17] J. Jing, A. S. Helal and A. Elmagarmid, Client-server computing in mobile environments, *ACM Computing Surveys*, vol.31, no.2, pp.117-157, 1999.

[18] C. Lee, C.-H. Ke and C.-C. Chen, Improving location management for mobile users with frequently visited locations, *Performance Evaluation Journal*, vol.43, no.1, pp.15-38, 2001.
[19] D. Milojicic, F. Douglis and R. Wheeler, *Mobility – Processes, Computers, and Agents*, Addison-Wesley, 1999.
[20] H. Nakano, A. Utani, A. Miyauchi and H. Yamamoto, Chaos synchronization-based data transmission scheme in multiple sink wireless sensor networks, *International Journal of Innovative Computing, Information and Control*, vol.7, no.4, pp.1983-1994, 2011.
[21] C.-H. Ou, Data delivery with mobility control in sparse wireless sensor networks, *International Journal of Innovative Computing, Information and Control*, vol.7, no.4, pp.1621-1638, 2011.
[22] E. Pitoura and G. Samaras, Locating objects in mobile computing, *IEEE Transactions on Knowledge and Data Engineering*, vol.13, no.4, pp.571-592, 2001.
[23] S. Papavassiliou, A. Puliafito, O. Tomarchio and J. Ye, Mobile agent-based approach for efficient network management and resource allocation: Framework and applications, *IEEE Journal on Selected Areas in Communications*, vol.20, no.4, pp.858-872, 2002.
[24] K.-L. Su, C.-Y. Chung, Y.-L. Liao and J.-H. Guo, A$*$ searching algorithm based path planning of mobile robots, *ICIC Express Letters, Part B: Applications*, vol.2, no.1, pp.273-280, 2011.
[25] V. S. Tseng, M.-H. Hsieh and K. W. Lin, A novel cache replacement algorithm for cooperative caching in wireless multimedia sensor networks, *International Journal of Innovative Computing, Information and Control*, vol.7, no.2, pp.763-776, 2011.
[26] C. Wang, X. Zheng, Y. Si and W. Liu, An enhanced-security fair payment protocol based on mobile agent, *ICIC Express Letters*, vol.5, no.3, pp.675-679, 2011.
[27] B. Zenel and D. Duchamp, A general purpose proxy filtering mechanism applied to the mobile environment, *Proc. of the 3rd Annual ACM/IEEE International Conference on Mobile Computing and Networking*, Budapest, pp.248-259, 1997.