

Research Article

Research on Migration Strategy of Mobile Agent in Wireless Sensor Networks

Dai Ting,^{1,2} Huang Haiping,^{1,2,3} Lu Yang,^{1,2} Wang Ruchuan,^{1,2,4} and Pan Xinxing^{1,2}

¹ College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

² Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, China

³ College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

⁴ Key Lab of Broadband Wireless Communication and Sensor Network Technology of Ministry of Education, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

Correspondence should be addressed to Huang Haiping; hhp@njupt.edu.cn

Received 16 July 2013; Accepted 9 September 2013

Academic Editor: Chongsheng Zhang

Copyright © 2013 Dai Ting et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Big data and distributed computing are of great importance in wireless sensor networks (WSNs). They are always bonded together, and the latter one upholds the former one. In distributed computing, mobile agent model is the mainstream technology. With autonomy, communicativeness, mobility, and role, mobile agent model is more suitable for large-scale, resources-restrained WSN to deal with big data. This paper mainly studies migration schemes for mobile agents, determines core factors of migration strategy, and proposes SMLA and DMLA algorithms. And aiming at revealing the characteristics of target tracking, this paper puts forward pid-DMLA algorithm; and considering multiple agents' cooperation, it presents Mpid-DMLA algorithm. Moreover, this paper evaluates and analyzes the advantages and disadvantages of the above-mentioned four algorithms by simulations.

1. Introduction

In wireless sensor networks (WSNs), big data is the intensification of information architecture, while distributed computing is the intensification of infrastructure and resource [1]. Distributed computing models can bring benefits to big data. And there are two categories of distributed computing: client/server model and mobile code model. Mobile agent model consists of remote computation, code on demand, and mobile agent model [2].

In client/server model, client asks for services, while server provides resources, services, and methods for executing services. There are many distributed systems adopting this model [3, 4]. Mainwaring proposed a habitat monitoring system, which is accomplished by transmitting data to gateway [3]. Ivy is the distributing system in WSN studied by the University of California, Berkeley, CA, USA [4]. Although the client/server model has been widely used, its drawbacks are very obvious, especially in WSN [5]. Firstly, it needs a series of supernodes as a processing center. Secondly, in multiclient systems, overdependence on bandwidth reduces

the transmitting performance. Thirdly, the nodes which are closer to the server need to transmit more data; thus, they consume more energy.

Mobile agent model supports distributed computing, too. The processing center/centralized information system (CIS) is in charge of sending and recovering mobile agents, while agents migrate between sensor nodes and are responsible for data exchange. A typical exemplar is the data fusion algorithm based on mobile agent to track objects in WSN [6]. In general, mobile agent is a special program which can be executed autonomously. And it is also defined as an entity with the following four attributes: agent identifier, data space, method, and migration table. Agent identifier describes the uniqueness of each agent; data space carries part of the integrated results; method is used to deal with missions or execute codes; and migration table stores the routing information of an agent and it can be preset or modified dynamically.

Compared with traditional computing models which need amount of data interaction between nodes and CIS (or base station), mobile agent model has many merits. As

illustrated above, in WSN, mobile agents dynamically migrate to the CIS with service request. With CIS accomplishing the computing operations locally and returning the results to the nodes, agents can be devoid of the transmission of big data and the reliance on the network bandwidth, which improves the efficiency. The specific advantages of mobile agent model, especially, in terms of energy consumption, reliability, and large-scale distribution, are declared in [7]. However, the consequent challenges are not negligible, like the high failure rate, unreliable communication link, high dependence on bandwidth, and so on.

2. Related Work

Although mobile-agent-based distributed model has exceptional advantages in WSN [8], dynamic migration is necessary in practical deployment. Dynamic migration, presented by migration table, has two behaviors: nodes selection and migration sequences. Therefore, migration table's efficiency, in some sense, decides accuracy of data integrations, time of migrations, energy consumption, and performance of the whole network.

In response to the challenges embodied in the specialty of WSN, the design of mobile agent's migration table needs to adequately consider the following three aspects. Firstly, agent entity must be as small as possible to reduce migrating energy consumption. Secondly, migration table has to guarantee rational precision of data integrations to obtain sufficient sensor nodes. Thirdly, in the precondition of the second aspect, migration table needs to be short enough. In conclusion, a suitable migration table should balance the cost and precision.

Thus, in this paper, we mainly concentrate on the design of the migration table which can reduce agent's migration energy consumption while guaranteeing reliable migration methods.

Briefly, mobile agent migration has two categories: static migration list algorithm (SMLA) and dynamic migration list algorithm (DMLA). In SMLA, CIS analyzes the whole network's resources and designs migration paths that are inserted in agent entities initially. However, static SMLA cannot respond to the changes of the network timely; the changes happen when one or more links between nodes break, while, in DMLA, migration table is updated at each migration interval according to the sensibility of the network.

Without loss of generality, mobile agent migration algorithms can be illustrated by target tracking. Considering movements of the target entity, Zhao et al. proposed the information-driven-based tracking method [9]. Liu et al. proposed the collaborative in-network method to track target [10], which effectively reduced overheads and energy consumption. Kim et al. transferred the distributed tracking issue to the continuous Bayesian estimation problem and adopted the information-driven-based sensor querying scheme to select the next hop [11]. Wang et al. proposed a heuristic target localization method based on entropy [12], which is more efficient than the information-based computing method. In this paper, we will propose the dynamic migration list

algorithm based on the predicting information drive (pid-DMLA) to solve the motion problems of migrations of the target entity.

Considering the condition of using multiple agents to track an application in practice, we will also propose the dynamic migration list algorithm based on the predicting information drive of multiple agents (Mpid-DMLA).

In this paper, we would propose the migration algorithms in different scenarios: single mobile agent and multiple mobile agents, which is valuable and applicable in WSNs. The rest of this paper is organized as follows. Section 3 describes the preliminary of migration. Section 4 presents single-agent-based migration algorithms, including SMLA, DMLA, and pid-DMLA. Section 5 illustrates Mpid-DMLA. Simulation results and analyses are presented in Section 6, and conclusions are drawn in Section 7.

3. Preliminary

3.1. Assumptions. The following assumptions are considered.

- (1) The location of each sensor node is known.
- (2) Each node has at least two neighbor nodes.
- (3) The valid perception radius of sensor node is more than half the distance between two neighboring nodes.
- (4) All nodes are synchronized.
- (5) All sensors are deployed in a two-dimensional region.
- (6) There is only one target entity in the region, and it moves uniformly and linearly.

3.2. Evaluation Indicators of the Agent Migration Algorithm. In order to evaluate the existing agent migration algorithms, we adopt the three indicators proposed by Xu et al. [7] energy consumption, network lifetime, and migration hops.

Each migration step of mobile agent is called a hop. And the energy consumption of each hop consists of three aspects: transmission consumption E_{ab} (E_{ab} represents the energy consumption of an agent migrating from node a to node b), initialization consumption E_i (E_i is constant), and data processing consumption. The network energy consumption is primarily concentrated in transmissions.

A migration is primarily composed of four parts: sending an agent, receiving an agent, data processing, and deciding the next hop. And each consumption time of the four processes is fixed. Thus, evaluating the migration time of an agent is equivalent to calculating the migration hops.

3.3. Node's Sensor Model. We adopt the following wireless transmission model [13] proposed by Rappaport:

$$z_k(t) \propto \frac{E}{\|X(t) - X_k\|^2}, \quad (1)$$

where $z_k(t)$ is a measure value of sensor k related to the location and time, E is the signal power of the target entity, $X(t)$ represents the location of the target entity in time

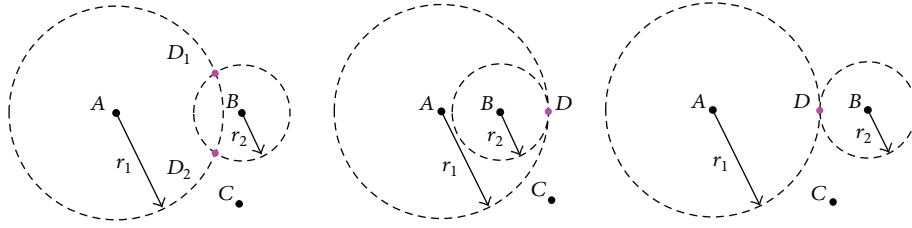


FIGURE 1: The target localization algorithm.

t , X_k represents the location of node k , $X(t)$ and X_k are two-dimensional vectors, and $\|X(t) - X_k\|$ represents the Euclidean distance between node k and target entity in time t .

3.4. Node's Information Collection Model. In order to quantify the data provided by sensors, we need to define the amount of information. According to (1), at any time, $z_k(t)$ is only related to $\|X(t) - X_k\|$. Thus, adopting the Gaussian model, the amount of information when the agent migrates to node k is the following:

$$I_k(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\|\bar{X}(t) - X_k\|^2 / 2\sigma^2}, \quad (2)$$

where σ is the standard deviation of the amount of information, which indicates the decrease of speed of $I_k(t)$ when the distance between sensor node and target entity increases, and $\bar{X}(t)$ represents the location of the target entity, which is calculated by the target localization algorithm in Section 3.6.

3.5. Beacon Frames. We use beacons to termly exchange information of the neighboring nodes. A beacon frame consists of the following five fields: X_k represents the location of node k ; $e_k(t)$ represents the residual energy of node k in time t ; $z_k(t)$ represents the sensor measurement of node k in time t ; and the last two fields are sending time t_s and receiving time t_r . Beacon frames are used to provide information of location and measurement to the target localization algorithm and the agent migration algorithm. Moreover, they can be used to test the survivability of neighbor nodes.

3.6. Target Localization Algorithm. As illustrated in (2), the location of target entity is calculated by the target localization algorithm. At present, target localization algorithms mostly adopt the method of centroid weighting. And the basic idea is to make node's beacons take control of the centroid's coordinate and to reflect the general relations of the locations by using weighting factors which influence the positions of the nodes' centroids. And according to the RSSIs of the three sensors' beacons, it is easy to obtain the other node's location by the trilateration.

As shown in Figure 1, the locations of $A(x_A, y_A)$, $B(x_B, y_B)$, and $C(x_C, y_C)$ are known; and one of $D(x_D, y_D)$ needs to be tested. According to the RSSIs, we can calculate

the distance of AD , BD , and CD , denoted by r_1 , r_2 , and r_3 , respectively. Thus, we can obtain the following equation set:

$$\begin{aligned} (x_B - x_D)^2 + (y_B - y_D)^2 &= r_2^2, \\ (x_A - x_D)^2 + (y_A - y_D)^2 &= r_1^2. \end{aligned} \quad (3)$$

Apparently, the distance of A and B is less than or equal to $r_1 + r_2$. So (3) has two real solutions that are identical or not, which correspond to the coordinates of the two nodes D_1 and D_2 that are identical or not. Select the one that is nearer to C as the approximate location of D , denoted as (x_1, y_1) . In the same way, we can obtain the other two approximate locations of D , denoted as (x_2, y_2) and (x_3, y_3) . Therefore, the coordinate of D is (x_D, y_D) , which can be calculated by the following:

$$\begin{aligned} x_D &= \frac{x_1/(r_1 + r_2) + x_2/(r_2 + r_3) + x_3/(r_3 + r_1)}{1/(r_1 + r_2) + 1/(r_2 + r_3) + 1/(r_3 + r_1)}, \\ y_D &= \frac{y_1/(r_1 + r_2) + y_2/(r_2 + r_3) + y_3/(r_3 + r_1)}{1/(r_1 + r_2) + 1/(r_2 + r_3) + 1/(r_3 + r_1)}. \end{aligned} \quad (4)$$

As a result, according to the previous target localization algorithm, we can obtain $\hat{x}(t)$.

4. Migration Algorithms Based on Single Mobile Agent

This section would propose SMLA and DMLA algorithms from static and dynamic aspects, respectively. And it would then propose pid-DMLA based on disadvantages of SMLA and DMLA.

4.1. SMLA. In SMLA algorithm, we use static information. Before distributing mobile agents, the CIS collects by communicating with other nodes, and it then generates a migration plan. In this migration plan, information collection values of all nodes in the network decrease. The specific steps of SMLA are as follows.

Algorithm 1 (SMLA).

- (1) At time $t = 0$, the node that has first detected information of target entity is denoted as cluster head; it generates a mobile agent.
- (2) At time $t = 0$, calculate location information of target entity, $\hat{X}(0)$, according to the target location algorithm.

Calculate the amount of information collection, I_0 , according to (2), and set $SumArray = I_0$.

Select the next hop, according to the predetermined migration sequence.

- (3) At time t , when agent reaches node k , the location of target entity is $\widehat{X}(t)$ according to the target location algorithm.

Calculate the amount of information collection of node k , I_k , and update $SumArray = SumArray + I_k$.

If $SumArray$ is greater than the threshold, then go to step 4; otherwise, continue to migrate to the next hop in the migration sequence, update time $t = t + 1$, and then repeat step 3.

- (4) Return to cluster head to process data.

Though, sometimes, SMLA can gain the ideal migration sequence, there are many drawbacks in practical applications. Firstly, since SMLA uses static information, sensor's transmission distance must be large enough to ensure that CIS is within one-hop range. So we need to adopt the nodes with greater power in communication capabilities. And using these nodes will inevitably result in relatively higher energy consumption. Secondly, since SMLA is a static algorithm, the static information, which is acquired before migration, lack timeliness and cannot reflect the changes of the network in the process of migrating. For example, that some nodes suddenly lose efficacy can result in the failure of the deployment of SMLA [9]. Moreover, most of WSNs are highly dynamic, so SMLA has some limitations in WSNs.

4.2. DMLA. SMLA is a centralized algorithm, and it has already decided the migration sequence before distributing mobile agents. Because WSN is a dynamic and distributed network, static routing information exposes a lot of problems in practical applications. Thus, DMLA is proposed. DMLA dynamically selects the next hop in the process of migration based on characteristics of the network.

4.2.1. The Migration Model of Mobile Agent in DMLA. In DMLA, mobile agents decide the nodes that they need to migrate to. In the surviving neighbor nodes, not all of the nodes can provide information affecting the final result. Therefore, the purpose of mobile agent migration algorithm is to produce a set of optimal subsets to determine the optimum migration sequence.

In the target entity tracking, we construct rules to choose the next hop as follows: mobile agent, which is located in node k , always looks for the next hop with minimal overheads. As the selection of the next hop is a method of balancing overheads and profit, we define the overheads function, when mobile agent moves from node i to neighbor node j at time t , as follows:

$$C_{ij}(t) = a \frac{E_{ij}}{E_{\max}} + b \left(1 - \frac{I_j(t)}{I_{\max}} \right) + \sqrt{1-a-b} \left(1 - \frac{e_j(t)}{e_{\max}} \right). \quad (5)$$

Equation (5) consists of three parts including energy consumption, information collection, and residual energy, where E_{ij} represents transmission energy consumption when the agent migrates directly from node i to node j ; $I_j(t)$ represents the amount of information collection of node j at time t ; $e_j(t)$ represents the residual energy of node j ; E_{\max} represents the maximum energy consumption of migration between two nodes, which is determined by the maximum transmission distance; I_{\max} represents the maximum amount of information collection; e_{\max} represents the initial energy of sensor node and it is a constant value; and a and b are the balance coefficients: $0 \leq a$ and $b \leq 1$.

Define the transmission energy consumption of the agent migrating from node i to node j as follows:

$$E_{ij} = P_{ij} \cdot T_{\text{trans}} = \alpha P_{r\text{-th}} \cdot \|X_i - X_j\|^2 \cdot T_{\text{trans}}, \quad (6)$$

where P_{ij} represents the transmission power of sensor nodes; $\|X_i - X_j\|$ represents the distance between node i and node j ; T_{trans} , a constant, represents the time of transmitting the mobile agent; $\alpha = 4\pi^2/\lambda^2 G_1 G_2$; G_1, G_2 are all of the antenna gain factors (set them as 1); λ represents the signal wavelength; and $P_{r\text{-th}}$ represents the receiving threshold.

From (2), (5), and (6), we can gain

$$C_{ij}(t) = a \frac{\|X_i - X_j\|^2}{d_{\max}^2} + b \frac{\|\widehat{X}(t) - X_j\|^2}{dt_{\max}^2} + \sqrt{1-a-b} \left(1 - \frac{e_j(t)}{e_{\max}} \right), \quad (7)$$

where dt_{\max} represents the maximum possible distance between two nodes.

Equation (7) can be rewritten as follows:

$$j^* = \arg \min \left\{ a \frac{\|X_i - X_j\|^2}{d_{\max}^2} + b \frac{\|\widehat{X}(t) - X_j\|^2}{dt_{\max}^2} + \sqrt{1-a-b} \left(1 - \frac{e_j(t)}{e_{\max}} \right) \right\}, \quad j \in N_i, \quad (8)$$

where N_i represents a group of neighbors of node i ; node $j^* \in N_i$ represents the next hop; and $j^* \in N_i$ can ensure less energy consumption, stronger sensing capabilities, and more residual energy.

In summary, the total cost in the migration process is

$$\text{Cost} = \sum_{p=0}^{\text{HOP}_s-1} C_{i_p i_{p+1}}, \quad (9)$$

$$\text{s.t. } SumArray = \sum_{p=0}^{\text{HOP}_s} I_p > \text{threshold},$$

where p represents the number of nodes migrated through HOP_s represents the total number of migration hops. In the practical migration, once mobile agent gains enough information, and it will terminate migration and return to the CIS.

4.2.2. The Specific Migration Procedure of DMLA. The mobile agent migrates to the alive node which has not been reached and with the minimal migration overheads according to (7). In this way, we can gain a table close to the optimal migration. If all nodes have been reached, mobile agent will return the node that distributes it. And when the mobile agent has acquired enough information, it will return to the CIS.

Since each node periodically receives beacon frames from neighbor nodes, DMLA can always use the updated local variables to determine the next hop. As only using local variables, sensor nodes can avoid the node which has a greater transmission distance, so part of the transmission energy can be saved. In addition, when some nodes suddenly lose efficacy, the whole migration process will not make mistakes. So DMLA algorithm is a fault-tolerant algorithm. The specific migration steps of DMLA are as follows.

Algorithm 2 (DMLA).

(1) At time $t = 0$, the node that has first detected information of target entity is denoted as cluster head; it generates a mobile agent.

(2) At time $t = 0$, calculate location information of target entity, $\widehat{X}(0)$, according to the target location algorithm.

Calculate the amount of information collection, I_0 , according to (2), and set $SumArray = I_0$.

Select the next hop, according to (8).

(3) At time t , when agent reaches node k , the location of target entity is $\widehat{X}(t)$ according to the target location algorithm.

Calculate the amount of information collection of node k , I_k , and update $SumArray = SumArray + I_k$.

If $SumArray$ is greater than the threshold, then go to step 4; otherwise, continue to migrate to the next hop according to (8), update time $t = t + 1$, and then repeat step 3.

(4) Return to cluster head to process data.

4.3. pid-DMLA

4.3.1. The Proposition of pid-DMLA. In the cooperation processing applications, the movement of target entity is one of the important factors that affect the final result [14–19]. DMLA algorithm does not take the direction of the movement of target entity into account. So in some cases, it cannot select the optimum next hop. For example, in Figure 2, at time t , the mobile agent is located in node A. Nodes B and C are neighbors of A, and they are equidistant to A. According to DMLA, mobile agent calculates node B as the next hop. This is because node B is nearer to the target entity than node C. Then, at time $t + 1$, the mobile agent is located in node B. But the target entity moves to a position closer to node C. Thus, DMLA, on the contrary, selects the node which is farther from the target and has smaller amount of information collection and measurement value

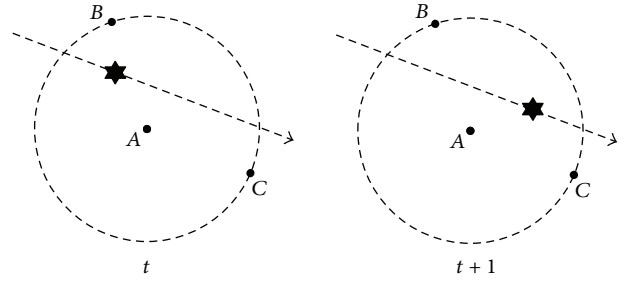


FIGURE 2: The movement of the target entity.

as the next hop. Therefore, when designing agent migration algorithms, we need to focus on predicting the movement of the target entity and selecting the node with the maximum information. And we will propose the pid-DMLA algorithm to solve the problems of the movement of the target entity in the migration process.

4.3.2. Agent Migration Model in pid-DMLA. Suppose that the moving direction and speed of the target entity do not change. Then, the change of the target entity's displacement is a constant, at each identical time interval, as follows:

$$X(t+1) - X(t) = X(t) - X(t-1). \quad (10)$$

Equivalently,

$$X(t+1) = 2X(t) - X(t-1). \quad (11)$$

As illustrated in (11), the position information $X(t+1)$ at time $t+1$ can be deduced from the positions at time $t-1$ and time t . In this way, the location information $\widehat{X}(t-1)$ and $\widehat{X}(t)$ can be calculated by target location algorithm; and, according to (11), the position of target entity, $\widehat{X}(t+1)$, can be estimated at time $t+1$ as follows:

$$\widehat{X}(t+1) = 2\widehat{X}(t) - \widehat{X}(t-1). \quad (12)$$

Update the overhead function in (7), and get

$$C_{ij}(t) = a \frac{\|X_i - X_j\|^2}{d_{\max}^2} + b \frac{\|\widehat{X}(t+1) - X_j\|^2}{dt_{\max}^2} + \sqrt{1-a-b} \left(1 - \frac{e_j(t)}{e_{\max}} \right), \quad (13)$$

where $\|\widehat{X}(t+1) - X_j\|$ represents the distance between node j and the target entity at time $t+1$.

And the next hop is

$$j^* = \arg \min \left\{ a \frac{\|X_i - X_j\|^2}{d_{\max}^2} + b \frac{\|\widehat{X}(t+1) - X_j\|^2}{dt_{\max}^2} + \sqrt{1-a-b} \left(1 - \frac{e_j(t)}{e_{\max}} \right) \right\}, \quad j \in N_i. \quad (14)$$

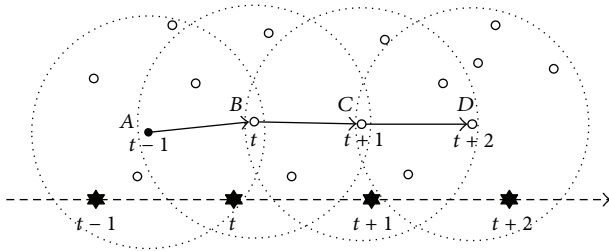


FIGURE 3: The migration of pid-DMLA.

At time $t = 0$, since the position of the target entity at the previous time is unknown, the amount of information collection can be calculated by the DMLA. At time $t > 0$, mobile agent migrates to the node closer to the estimated position of the target entity. So it can obtain larger sensing measurement and information collection, which is the core of pid-DMLA.

4.3.3. The Specific Migration Procedure of pid-DMLA. The process of mobile agent migration of pid-DMLA is shown in Figure 3. In a period of time, the target entity moves along the dotted line, and mobile agent migrates from node A to D through B and C. As shown in Figure 3, mobile agent always migrates to the node nearest to the next position of the target entity.

The specific migration procedure of pid-DMLA is as follows.

Algorithm 3 (pid-DMLA).

- (1) At time $t = 0$, the node that has first detected information of target entity is denoted as cluster head;

it generates a mobile agent.

- (2) At time $t = 0$, calculate location information of target entity, $\widehat{X}(0)$, according to the target location algorithm.

Calculate the amount of information collection, I_0 , according to (2), and set $SumArray = I_0$.

Select the next hop, according to (14).

- (3) At time t , when agent reaches node k , the location of target entity is $\widehat{X}(t)$ according to the target location algorithm.

Calculate the amount of information collection of node k , I_k , and update $SumArray = SumArray + I_k$.

If $SumArray$ is greater than the threshold, then go to step 4; otherwise, continue to migrate to the next hop according to (14), update time $t = t + 1$, and then repeat step 3.

- (4) Return to cluster head to process data.

5. Migration Algorithm Based on Multiple Mobile Agents

5.1. The Architecture of Mpid-DMLA. Many researchers have proposed and realized feasible ways to organize the migration of a single mobile agent, such as Agilla program [18] and SensorWare program [19]. Consider the functional requirements of these programs. Firstly, they have not formed an architecture to manage the agent migration mechanisms in WSN. Secondly, they lack migration algorithms when a tracking application needs the collaboration of multiple agents. So when dealing with multiple agents' migration, we need a specific agent to manage the applications.

Therefore, we will propose Mpid-DMLA to solve this problem.

Suppose that multiple target entity applications are deployed in an area of sensors; each application is responsible for monitoring a part of the area and composed of one or more mobile agents. Therefore we have the following definitions.

Definition 4 (code segment application). A centralized program which consists of multiple mobile agents deployed in the WSN to accomplish a specific task is called centralized code segment application of multiple agents, which is called code segment application for short.

Code segment applications are associated with sensing data that need to be evaluated, and they need to cooperate and interact based on the need of programs. On the other hand, according to the Deluge & Mate model [20], code segment applications need to redeploy themselves so that they can be distributed by the CIS.

To track targets, code segment applications require the following four functions [21].

- (1) *Sensing.* Sensor nodes are distributed around the target node, and they test and collect information of the surrounding environment.
- (2) *Processing Information.* After acquiring sensor information, most of the nodes send it to the CIS for processing, while cluster heads will directly process information.
- (3) *Communication.* In order to maintain the application, sensor nodes need to keep exchanging information with other nodes or CIS.
- (4) *Migration.* Mobile agents migrate based on the need of code segment application.

Constructing the architecture of multiple mobile agents also requires the following definitions.

Definition 5 (main component agent). The main component agent is responsible for managing migration information and activity cycle of all component agents, processing sensing data of other agents, and collecting and exchanging the information of subcomponent agents. Each code segment application has one and only main component agent.

Definition 6 (subcomponent agent). The subcomponent agent is used to obtain the data of nodes and report them to the main component agent. Each code segment application has one or more subcomponent agents.

Definition 7 (communication component agent). The communication component agent is used to collect and exchange the entity information of WSN by migrating. Each code segment application has one or more communication component agents.

Definition 8 (middleware software). The middleware software, located between the physical and the operating system layers, is the software entity running in mobile agents.

5.2. *The Migration Model of Mobile Agent in Mpid-DMLA.* When a code segment application changes monitoring positions, all of the mobile agents in this application need to migrate together to reconstruct a new code segment application. In Agilla, the main component agent controls the entire process by sending the migration sequence, while other mobile agents can communicate by accessing the independent tuple space [22]. And the engine virtual machine works as the kernel to provide services and controls all of the concurrent execution of all of the agents in a node. However, with the increase of the mobile agents, the possibilities of the loss of packet and the failure of migration will increase, too. Suppose that a code segment application is composed of N mobile agents. Define P as the failure coefficient of a single agent. And then we can obtain the success coefficient of migration, in the entire application, as follows:

$$P_{\text{success}} = (1 - P)^N. \quad (15)$$

Communication overheads, based on pid-DMLA, are given as follows:

$$C_{ij}(t) = \left[a \frac{\|X_i - X_j\|^2}{d_{\max}^2} + b \frac{\|\widehat{X}(t+1) - X_j\|^2}{dt_{\max}^2} + \sqrt{1 - a - b} \left(1 - \frac{e_j(t)}{e_{\max}} \right) \right] \times (P_{\text{success}})^{-1}. \quad (16)$$

And the next hop is

$$j^* = \arg \min \left\{ \left[a \frac{\|X_i - X_j\|^2}{d_{\max}^2} + b \frac{\|\widehat{X}(t+1) - X_j\|^2}{dt_{\max}^2} + \sqrt{1 - a - b} \left(1 - \frac{e_j(t)}{e_{\max}} \right) \right] \times (P_{\text{success}})^{-1} \right\}. \quad (17)$$

5.3. *The Specific Migration Procedure of Mpid-DMLA.* In the construction of the target entity tracking, we assume that subcomponent agent is a small program of lightweight code which is suitable for migration. And it can independently

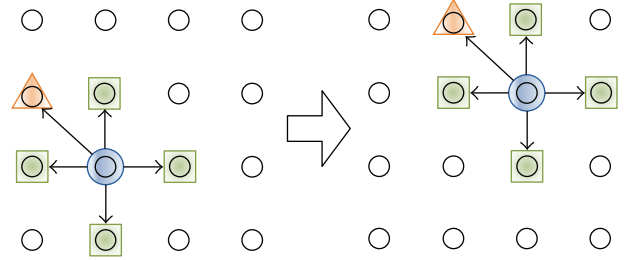


FIGURE 4: Migration based on predicting information drive of multiple mobile agents.

access the target entity's location. As shown in Figure 4, hollow circles represent sensor nodes which are distributed in the sensing region; code segment is composed of one main component agent (solid circle), four subcomponent agents (solid squares) and one communication component agent (solid triangle). Each agent runs on one node, executing code segment applications. The migration of code segment application means the migration of the six mobile agents, which is the migration of multiple mobile agents [19]. The following algorithm can be used to describe the above-mentioned procedure.

Algorithm 9 (Mpid-DMLA).

- (1) A main component agent, a communication component agent, and several subcomponent agents are deployed in different sensor nodes.
- (2) Sensor nodes are monitoring around the target entity.

The node that has first detected information of target entity is denoted as cluster head; the main component agent then migrates to it.

- (3) At time $t = 0$, calculate location information of target entity, $\widehat{X}(0)$, according to the target location algorithm.

Calculate the amount of information collection, I_0 , according to (2).

Define the threshold of the amount of information as *Threshold_UP*, and let *Threshold_UP* = I_0 .

The main component agent generates new communication component agent and transfers I_0 and *Threshold_UP* to it.

Then, the communication component agent sets to migrate.

- (4) At time t , when agent reaches node k , the location of target entity is $\widehat{X}(t)$ according to the target location algorithm.

Calculate the amount of information collection of node k , I_k , and update *Threshold_UP* = *Threshold_UP* + I_k .

If *Threshold_UP* is greater than the threshold, then go to step 5; otherwise, select the next hop for migration according to (17), update time $t = t + 1$, and then repeat step 3.

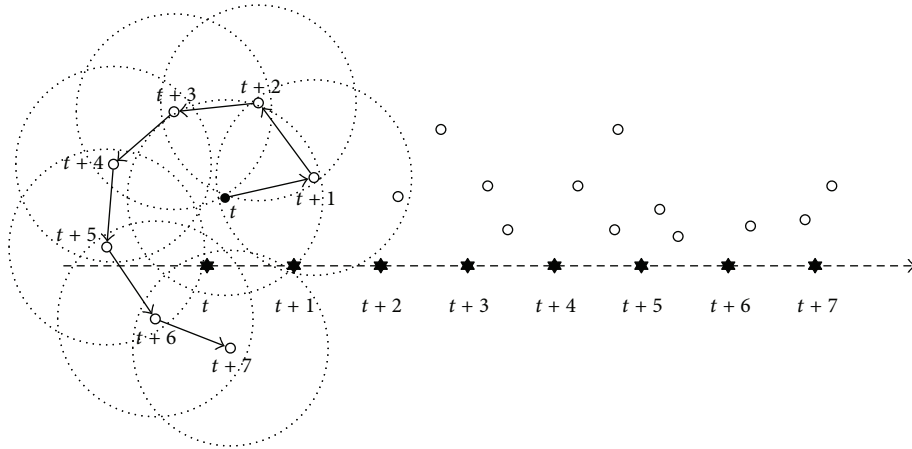


FIGURE 5: The migration process of SMLA.

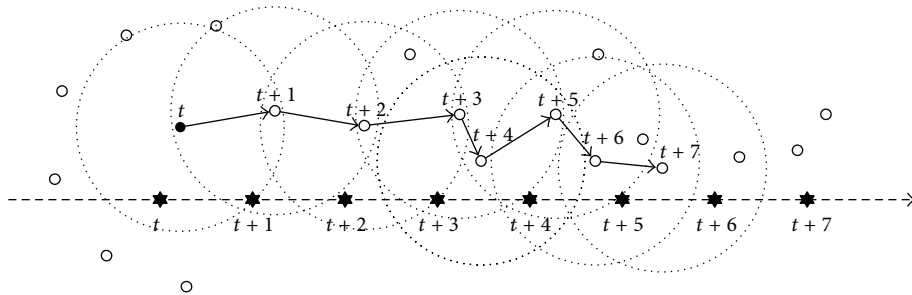


FIGURE 6: The migration process of DMLA.

- (5) Communication component agent stores the information of the migration path: $(\{I_0(t=0), I_1(t=1), \dots, I_k(t=k)\}, \{X(t=0), X(t=1), \dots, X(t=k)\})$ and other relevant information.

And it interacts with the main component agent.

- (6) The main component agent generates the migration sequence $\{N_1, N_2, \dots, N_k\}$, according to location of target entity.

Send it to each subcomponent agent (this step is optional for different applications).

- (7) The main component agent sends complete instructions to the lower-layered middleware software.

Method functions which are predefined in the middleware software recycle the communication component agent and the subcomponent agents.

- (8) The main component agent goes to the destination node based on the migration sequence.

If the migration fails, send failure information to CIS to terminate this migration; otherwise, generate new communication component agent and subcomponent agents, and this migration ends.

6. Simulation and Analyses

6.1. The Effect of Single Mobile Agent in Target Tracking. Before the simulation experiments, we firstly use target tracking as a sample to demonstrate the migration effect of a single mobile agent of different migration algorithms. As shown in Figures 5, 6, and 7, suppose that a car is running from left to right through the area covered by WSN. The small circle represents the sensor node; solid small circle represents the cluster head, which distributes agents; large dotted circle represents the communication range of sensor node, and each node selects the next hop only within its communication range. The above-mentioned figures illustrate the specific locations of the car and agent from time t to time $t+7$.

The migration process of SMLA is shown in Figure 5. At time t , CIS distributes the mobile agent to the node which is the nearest to the target entity, according to the predetermined migration sequence. When the object is stationary or its motion can be estimated, this static approach is very suitable. However, it loses efficacy in some conditions. For example, if the object remains in the location at time $t+1$, then the migration sequence will be the positions that are around the target entity. But once the target entity starts to move not according to the predetermination, SMLA will fail.

DMLA can, in some sense, optimize the performance of SMLA. As shown in Figure 6, mobile agent tracks the movement of the target entity. However, the movement of

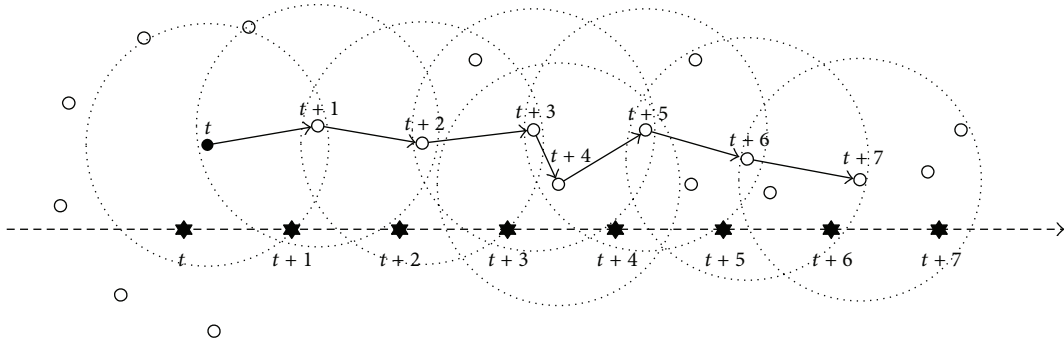


FIGURE 7: The migration process of pid-DMLA.

TABLE 1: Basic simulation parameters of single mobile agent migration algorithms.

Network area	500 × 500 m
Number of nodes	200 or 800
Node's initial energy	50 J
Node's sensing radius	5 m
Energy consumption for data processing	0.3 J
Energy consumption for sending beacon frames	0.1 J
The interval of node broadcasting beacon frames	0.1 s
The speed of the target entity	0~50 m/s
The threshold of the amount of information	18
The size of a mobile agent	800 bits
Simulation time	100 s

the target entity is unpredictable, so it is not accurate to decide migration path only by the amount of information, which is, however, related to the distance to the target entity.

The pid-DMLA can predict movement information of target entity dynamically, which is a better solution to such problems. As shown in Figure 7, at time $t + 5$, mobile agent selects the next hop according to the predicted information, which is different from DMLA. In brief, pid-DMLA rebuilds migration sequence based on predicting the movement of the target entity to achieve the purpose of tracking.

6.2. Performance Analysis on the Single Mobile Agent Migration Algorithms. We use simulation software, NS2, to evaluate the three single mobile agent migration algorithms' performances. The default simulation parameters are shown in Table 1.

Suppose that once the agent migrates one hop, all sensor nodes in the network consume 0.2 J. For SMLA, the transmission distance is set to 30 m, while, for DMLA and pid-DMLA, the transmission distance is set to be 10 m. Therefore, the consumption of transmission, for SMLA, is three times more than that of sending. The reason for this setting is that SMLA needs bigger transmission range to collect information. The direction and size of movement of the target entity are constant, denoted as v (m/s). Once the target entity reaches the edge of the sensing area, it reverses direction immediately, which ensures its being always within the sensing area.

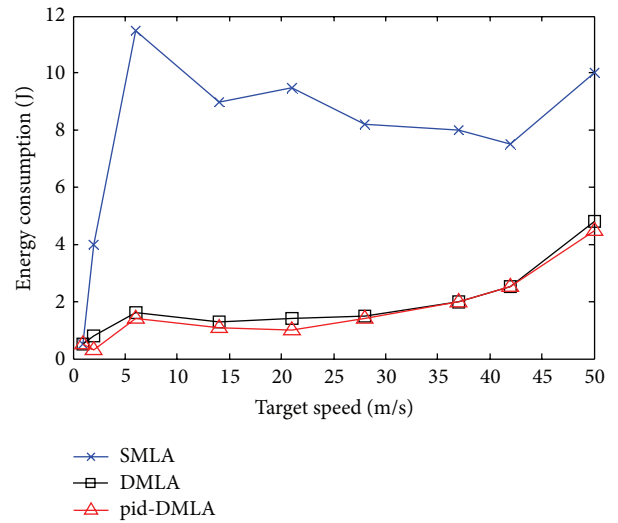


FIGURE 8: Energy consumption of the single agent migration algorithms in Scenario 1.

Scenario 1. Set the number of nodes as 200, the target entity speed as 0~50 m/s, and the simulation time as 100 s, which can guarantee a complete migration. The simulation results are shown in Figures 8 and 9. It is obvious that DMLA and pid-DMLA consume less energy than SMLA in the whole network. This is because the communication cost of SMLA is higher than that of DMLA. But with the increasing speed of the target entity, the effect of SMLA algorithm in tracking the target entity becomes less and less obvious; therefore, migration may terminate, and the energy consumption, on the contrary, becomes slower. On the other hand, when the speed of target entity is very low, there are no significant differences among the three migration schemes.

Scenario 2. Set the speed of target entity as 5 m/s and the number of nodes as 800. Generally, the coverage and the number of nodes are positively correlated. The simulation results are shown in Figures 10 and 11. It is clear that pid-DMLA has the lowest energy consumption when the number of nodes is more than 100. This is because with the increase of the nodes the choices of predicting the next hop become

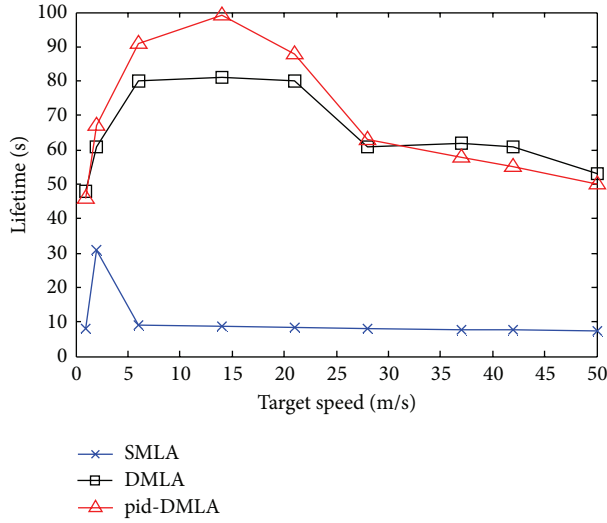


FIGURE 9: Lifetime of the single agent migration algorithms in Scenario 1.

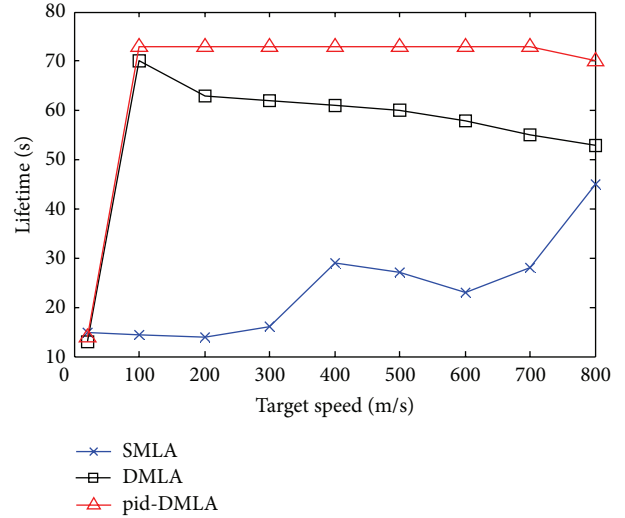


FIGURE 11: Lifetime of the single agent migration algorithms in Scenario 2.

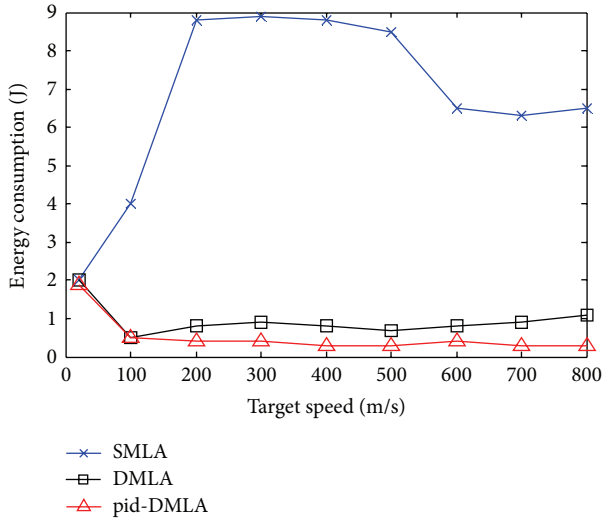


FIGURE 10: Energy consumption of the single agent migration algorithms in Scenario 2.

more; as a result, the overheads become less [23]. When the number of nodes is less than 50, the performance of SMLA is better than those of the other two. This is because with less nodes static and dynamic migration algorithms share the more similar migration path; however, the migration table has been predetermined in SMLA; thus, it, on the contrary, has the least overheads. On the other hand, when the density of nodes is high, there would be more neighbors and more migration choices. Thus, DMLA and pid-DMLA outperform SMLA.

6.3. Multiple Mobile Agents Migration Algorithm Performance Analysis. Before the assessment of Mpid-DMLA, we need to consider two problems [24, 25]. The first is the effect of code

segment applications on the amount of communication in the migration process. To ensure the comprehensiveness of the evaluation, set the number of communication component agents as 1 and the number of subcomponent agents as 5, 10, or 20, and evaluate Mpid-DMLA based on code segments with different scales. The second is the effect of different environment and running platform in practical operation on the amount of information and the success rate of migration. Therefore, we only consider the relation of the number of communication and energy instead of the influence of the agent's size.

The performance of the migration of mobile agent in Agilla system has been evaluated in [19]. First, Agilla system is not designed to support multiple mobile agents specifically, which means that it is not aware of the number of the agents in the code segment application. And the system's overheads are related to the size of mobile agent. So, in Agilla, all kinds of agents are considered to be identical, and there is no discrimination between the main and communication component agents. However, this does not affect the simulation results because all agents are uniformly assumed in the simulation in this paper.

The purpose of multiple mobile agents migration is to complete the entire application migration with limited energy. The default simulation parameters are shown in Table 2.

And when main component agent recycles and sends mobile agents, suppose that the energy consumption of the node where main component agent stays is neglected, while that of other nodes is 0.05 J.

Figures 12, 13, and 14 illustrate the relation between the number of application code segments and the success rate of one hop migration, when the numbers of subcomponent agents are 5, 10, and 15, respectively. The success rate of Mpid-DMLA is higher, when the number of subcomponent agents is 5, than those when the numbers of subcomponent agents

TABLE 2: Basic simulation algorithm parameters of a single mobile agent migration.

Network area	100 × 100 m
Number of nodes	7 × 7
Number of main component agents per code segment application	1
Number of subcomponent agents per code segment application	5, 10, or 15
Number of communication component agents per code segment application	1
Node's initial energy	2 J
Energy consumption of a node when an agent reaches it	0.05 J
Interaction consumption when an agent migrates	0.1 J

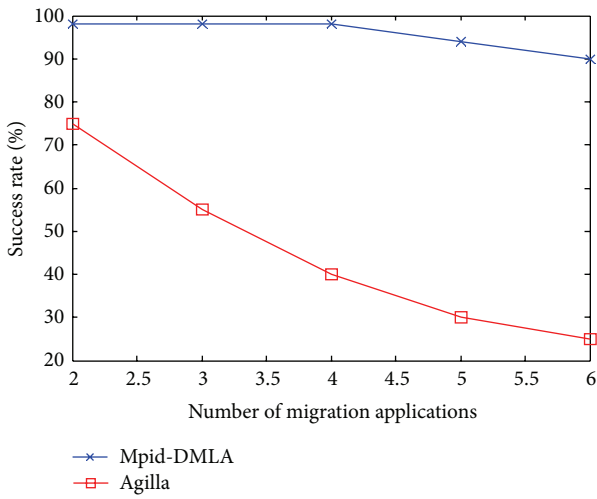


FIGURE 12: The relation between the number of code segment applications and the success rate of migration (I).

are 10 and 20. It is obvious that the successful rate of Mpid-DMLA is generally not less than 50%, which is much larger than that of Agilla.

Moreover, Figure 15 demonstrates the relationship between the migration hops and the amount of communication, when the subcomponent agents are 5. As shown in Figure 15, with the same migration hops, the amount of communication of Mpid-DMLA is much less than that of Agilla. As the number of data packet reflects the amount of energy consumption, Mpid-DMLA consumes less energy than Agilla.

Although the simulation results show that Mpid-DMLA is a suitable migration scheme for multiple mobile agents, it bases on some specific assumptions. In general, this scheme still encounters the following three issues.

- (1) When the main component agent loses efficacy in the process of migration, code segment applications cannot be rebuilt.
- (2) When subcomponent agents lose efficacy after deployment, code segment applications will keep waiting because the applications cannot be recycled.

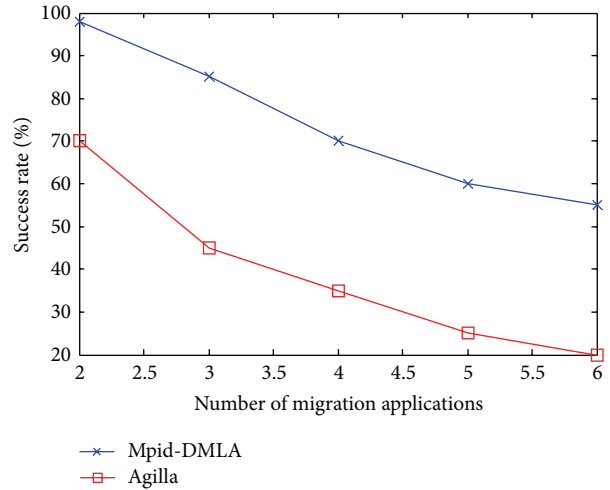


FIGURE 13: The relation between the number of code segment applications and the success rate of migration (II).

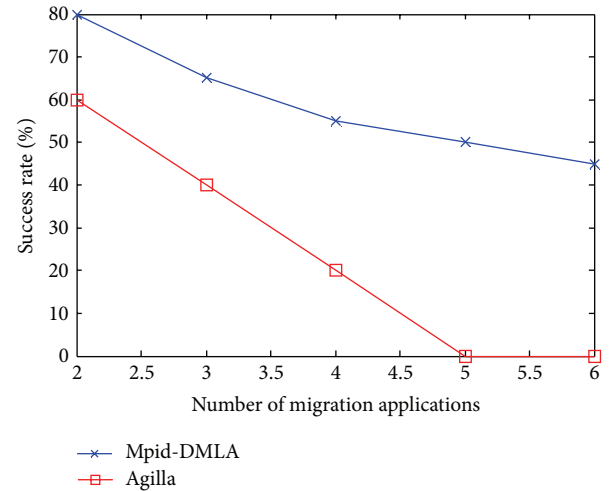


FIGURE 14: The relation between the number of code segment applications and the success rate of migration (III).

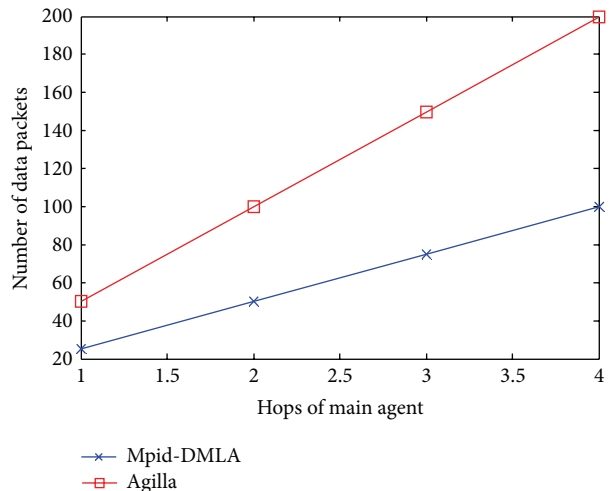


FIGURE 15: The relation between the migration hops and the amount of communication.

- (3) If the main and the subcomponent agents, which lose efficacy in issue 1 or 2, are left in the nodes without being recycled, then other mobile agents cannot reach these nodes.

Therefore, in designing middleware software, it is necessary to add the feedback mechanism of main component agent, and it needs to periodically return ACK information to the higher layer of the middleware software. When the middleware finds that the main component agent loses efficacy, it needs to distribute a new one or a specialized independent agent to recycle the useless agents in the whole network.

In Agilla, the method of recycling the useless agents is by adding the unified interface into all of the agents. Removing all of the mobile agents will affect the function of code segment applications. Therefore, developing the more appropriate middleware software is the future work, which is also related to the development of the better migration strategies [26].

7. Conclusion

As big data is the intensification of information architecture while distributed computing is the intensification of infrastructure and resource, distributed computing is an appropriate method to deal with big data. In this paper, we have firstly illustrated the mobile agent model in the distributed computing. And it led to the two considerable factors of migration: the selection of nodes and of the sequence of them. Then, we have proposed the assumptions of the migration algorithms. Moreover, we have designed the node's sensor model and information collection model. Furthermore, we have formulated the beacon frames of mobile agent for communication. Generally, mobile agent migration algorithms can be transferred as the target tracking issue. Thus, we have described a simple target localization algorithm applicable for migration. On this basis, we have generalized the whole migration process of the mobile agent.

Considering the single mobile agent, we have proposed the static SMLA and the dynamic DMLA algorithms. After analyzing the drawbacks of SMLA and DMLA, we then proposed the pid-DMLA algorithm. Section 4 has demonstrated the modeling procedure of the abovementioned three algorithms; Section 6 has emulated and evaluated their performances and studied their advantages and disadvantages in different scenarios.

Considering the multiple mobile agents applied in practical target tracking, we have also put forward the Mpid-DMLA algorithm. In the process of mobile agent component constructing code segment applications, we defined three mobile agents with different functions to cooperate with each other to accomplish the migration. Section 5 illustrated the architecture, modeling procedure, and basic process of Mpid-DMLA; Section 6 compared it with Agilla and evaluated its advantages and disadvantages.

Acknowledgments

The paper is sponsored by the National Natural Science Foundation of China (61170065, 61171053, 61003039, 61100199, 61003236, 61202355, 61201163, and 61373138), the Natural Science Foundation of Jiangsu Province (BK2011755, BK2011072, and BK2012436), the Scientific & Technological Support Project of Jiangsu Province (BE2012183 and BE2012755), the Natural Science Key Fund for Colleges and Universities in Jiangsu Province (12KJA520002), the Scientific Research & Industry Promotion Project for Higher Education Institutions (JHB2012-7), the Postdoctoral Foundation (2012M511753, 1101011B, 1202034C, and 2013T60536), the Doctoral Fund of Ministry of Education of China (20113223110002 and 20123223120006), the Fund of Nanjing University of Posts and Telecommunications (NY212047), and the project funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (yx002001).

References

- [1] Research Trends special issue on big data, http://www.research-trends.com/wp-content/uploads/2012/09/Research_Trends_Is-sue30.pdf, 2012.
- [2] L. Ismail and D. Hagimont, "A performance evaluation of the mobile agent paradigm," *ACM SIGPLAN Notices*, vol. 34, no. 10, pp. 306–313, 1999.
- [3] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 88–97, September 2002.
- [4] IVY-A sensor network infrastructure for the college of engineering, <http://www-bsac.eecs.berkeley.edu/projects/ivy>, 2008.
- [5] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–105, 2002.
- [6] Y.-C. Tseng, S.-P. Kuo, H.-W. Lee, and C.-F. Huang, "Location tracking in a wireless sensor network by mobile agents and its data fusion strategies," *Computer Journal*, vol. 47, no. 4, pp. 448–460, 2004.
- [7] Y. Xu and H. Qi, "Mobile agent migration modeling and design for target tracking in wireless sensor networks," *Ad Hoc Networks*, vol. 6, no. 1, pp. 1–16, 2008.
- [8] Y. Xu and H. Qi, "Distributed computing paradigms for collaborative signal and information processing in sensor networks," *Journal of Parallel and Distributed Computing*, vol. 64, no. 8, pp. 945–959, 2004.
- [9] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 61–72, 2002.
- [10] J. Liu, J. Reich, and F. Zhao, "Collaborative in-network processing for target tracking," *Eurasip Journal on Applied Signal Processing*, vol. 2003, no. 4, pp. 378–391, 2003.
- [11] J.-W. Kim, J.-S. In, K. Hur, J.-W. Kim, and D.-S. Eom, "An intelligent agent-based routing structure for mobile sinks in WSNs," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 4, pp. 2310–2316, 2010.
- [12] H. Wang, G. Pottie, K. Yao, and D. Estrin, "Entropy-based sensor selection heuristic for target localization," in *Proceedings*

- of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04), pp. 36–45, ACM Press, April 2004.
- [13] T. S. Rappaport, *Wireless Communications: Principles and Practice*, IEEE Press, 1996.
- [14] H. Mousannif and I. Khalil, “Cooperative data management in wireless sensor networks,” *International Journal of Computational Intelligence Systems*, vol. 5, no. 3, pp. 403–412, 2012.
- [15] N. Labraoui, M. Guerroui, M. Aliouat, and T. Zia, “Data aggregation security challenge in wireless sensor networks: a survey,” *Ad-Hoc and Sensor Wireless Networks*, vol. 12, no. 3-4, pp. 295–324, 2011.
- [16] M. Chen, L. T. Yang, T. Kwon, L. Zhou, and M. Jo, “Itinerary planning for energy-efficient agent communications in wireless sensor networks,” *IEEE Transactions on Vehicular Technology*, vol. 60, no. 7, pp. 3290–3299, 2011.
- [17] Q. Wu, N. S. V. Rao, J. Barhen et al., “On computing mobile agent routes for data fusion in distributed sensor networks,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 6, pp. 740–753, 2004.
- [18] C.-L. Fok, G.-C. Roman, and C. Lu, “Mobile agent middleware for sensor networks: an application case study,” in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 382–387, April 2005.
- [19] A. Boulis, C. Han, and M. B. Srivastava, “Design and implementation of a framework for efficient and programmable sensor networks,” in *Proceedings of the 1st international conference on Mobile systems, applications and services*, ACM, San Francisco, Calif, USA, 2003.
- [20] P. Levis and D. Culler, “Mate: a tiny virtual machine for sensor networks,” *ACM Sigplan Notices*, vol. 37, no. 10, pp. 85–95, 2002.
- [21] S. Suenaga, N. Yoshioka, and S. Honiden, “Group migration by mobile agents in wireless sensor networks,” *Computer Journal*, vol. 54, no. 3, pp. 345–355, 2011.
- [22] P. Bellavista, Y. Cai, and T. Magedanz, “Recent advances in mobile middleware for wireless systems and services,” *Mobile Networks and Applications*, vol. 16, no. 3, pp. 267–269, 2011.
- [23] Z. Zhirou and C. Wengang, “Model and optimization of mobile agent’s migration in grid,” in *Proceedings of the 2nd International Conference on Bio-Inspired Computing: Theories and Applications (BICTA '07)*, pp. 127–130, September 2007.
- [24] J.-P. Zhang, M. Jun, J. Yang, and L.-L. Cheng, “An improved migration strategy of mobile agent,” in *Proceedings of the 4th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD '07)*, pp. 577–581, August 2007.
- [25] Y. Lee and K. Kim, “Optimal migration path searching using path adjustment and reassignment for mobile agent,” in *Proceedings of the 4th International Conference on Networked Computing and Advanced Information Management (NCM '08)*, pp. 564–569, Gyeongju, South Korea, September 2008.
- [26] M. Tentori, M. Rodríguez, and J. Favela Vara, “An agent-based middleware for the design of activity-aware applications,” *IEEE Intelligent Systems*, vol. 26, no. 3, pp. 15–23, 2011.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

