

Multiple Mobile Agents for Data Collection in Overflow Time Based Partitions of Wireless Sensor Networks with Dynamic Deadlines

Pranesh V Kallapur, Niranjan N Chiplunkar

Department of Computer Science, N.M.A.M Institute of Technology, Nitte, Karnataka, India
praneshvkallapur@yahoo.co.in, nchiplunkar@nitte.ac.in

Abstract

Recent research has provided means by which wireless sensor network with dynamic deadlines can take advantage of mobile agents for the purpose of data collection to make efficient use of energy. Already lot of research has been done in this direction in case of wireless sensor networks both of static & dynamic type using single mobile agent. It may happen to be the case that, as network size increases, single mobile agent may not be able to meet deadline requirements of entire network. Consequently there may be occurrence of large number of buffer overflows leading to data loss. This paper proposes a technique to partition given wireless sensor network into two parts based on overflow time threshold and use a mobile agent in each partition for data collection. Also, we give analysis of results of applying the proposed technique on wireless sensor networks of varying sizes with randomly assigned overflow times which shows that number of buffer overflows can be drastically reduced by applying technique proposed in this paper. Finally, paper will be concluded with mentioning of future directions of research.

Keywords – Topology, Wireless Sensor Networks, Scheduling, partitioning, and multiple mobile agents

1. Introduction

Nowadays sensor networks are frequently being used to sense and measure the environment. Different protocols have been suggested for data transfer task by sensor nodes to Processing Element. Efficient use of energy has still been one of the issues of critical importance. The other issue related to energy consumption is non-uniform variation of phenomenon in different regions in the network. To capture this behavior the sensing rates of sensors at different positions will typically need to be different. The sensor nodes in regions with higher rate of variation in the phenomenon need to sample more frequently. This causes some sensor nodes accumulating enough information at a faster rate than other nodes causing buffer overflow & overflow to be dynamic in nature [1]. As a consequence of this, such nodes need to transfer their data to Processing Element more frequently than others. This causes nodes to run out of their batteries making entire network to reach to a useless state. Mobility has been proposed as a solution to this problem.

Mobile agents can traverse the network and collect data from sensor nodes when they come near sensor nodes. References [2, 3, 5, 6] discuss about using existing mobility or mobile agents added to the system explicitly as discussed in [4, 7, 8]. The path along which the mobile agent moves can be predefined and was discussed in [4]. Mobile agent, while

visiting a sensor node, receives the data from sensor node and stores it in its own memory which frees the buffer of sensor node. The mobile agent continues this task of data gathering as long as its battery level is above minimum threshold value. Then mobile agent returns to Processing Element to transfer collected data and can get recharged to start next round of data gathering. The issue in using mobile agent for data gathering is determining the sequence of visits to sensor nodes so that none of the sensor nodes' buffer overflows. This is called as **Mobile Element Scheduling (MES)** problem, which is discussed in detail, formulated as ILP, and proven to be NP-Complete in [1].

Earlier research works, those mentioned in section 2 and including those proposed in [1], try to make use of a single mobile agent to meet dynamic deadline requirements of entire topology irrespective of its size in terms of number nodes in it. But, a single mobile agent may not be able to meet, even after applying algorithms as proposed in [1] which aim for blending cost and deadline proportionately, the dynamic deadlines of entire network. This happens due to the fact that, in practice, applications of wireless sensor networks will normally be spread over large geographical areas. There will be large number of buffer overflows in case of such wide area sensor networks. This may be due to either a smaller subset of very short deadlines or locally minimal wireless links from the point of view of cost, resulting in mobile agent stuck between servicing only small subset of sensor nodes in entire network. This situation may make other sensor nodes to wait for longer time, than their buffer capacities, in order to transfer data they have in their buffer to mobile agent. This kind of longer waiting by sensor nodes beyond their stipulated buffer capacity increases the number of buffer overflows. These buffer overflows leads to data loss which directly means less accurate sensing of parameters in which the wireless network is interested.

Hence, use of single mobile agent for networks with large number of sensor nodes may be infeasible from the point of view of number of buffer overflows and accuracy of data collection.

Thus, motivation of this paper is to propose a technique to reduce the total number of buffer overflows in a wireless sensor network with dynamic deadlines, as much possible, helping more accurate data collection.

The rest of this paper is organized as follows: Section 2 discusses related work. In section 3 discusses in detail about the technique proposed by this paper. Section 4 shows simulation results of proposed algorithm. Section 5 discusses about the results presented and paper is concluded in Section 6 outlining some directions for future work.

2. Related Work

References [7, 8] discuss about using a message ferry for routing data from one node to another in case of sparse network. In particular, [7] gives schemes to find the route ferry has to take. Based on a given traffic matrix (expected traffic from any node to any other node), goal is to find the optimal route of ferry so that average delay from source to destination is minimized, meeting the bandwidth requirements of the traffic based on the bandwidth requirements of given network.

Vehicle Routing Problem (VRP) proposed in [9] has some similar approach to that of using a mobile agent. In VRP, a set of nodes need to be visited, but there can be more than one vehicle. Vehicles start and end at a special node (*node₀*), which is referred to as the depot. Although work proposed in [9] looks similar to the one being presented in this paper, actually it is different for two reasons. First reason is that, as given in [9], it involves use of multiple vehicles over entire network without partitioning it. Second difference is that tour predicted by [9] will be based on Hamiltonian tour assigned to each vehicle such that sum of the distances travelled by each vehicle is minimum. But, the nature and goal of tour involved in the problem presented here are different. Nature of tour involved is repetitive and dynamic and goal is to reduce the number of buffer overflows which occur periodically. Hence VRP problem and problem addressed in this paper are different.

There are many variants to VRP. Those similar to our problem are VRP with time Windows (VRPTW), and Periodic VRP (PVRP). In VRPTW [10, 11], in addition to the VRP constraints, there is a time window within which each node has to be visited. But both are different from the problem discussed here, as in our case, before visiting all the nodes once, some node (which is sampling at a higher rate) may need to be visited more than once. In addition, the time windows of the visits are not known apriori. Only the current time window is known, and the next time window is decided based on the current visit time. In PVRP [11], each node has to be visited a pre-specified number of times, but there are no time window constraints on the visits.

Optimizing speed and energy of mobile agents have been proposed in [15, 16] but only with respect to wireless sensor networks with periodic time windows using a single mobile agent and not for dynamic deadlines. Some work has been proposed for splitting network into partitions in [14] called Partition Based Scheduling (PBS), but effectively uses a single mobile agent to collect data finally, which it claims to 'prevent' buffer overflow from any sensor node. But, mobile

agent scheduling has been already proven to be NP-Complete in [1].

Few heuristic based algorithms have been proposed in [1] to address issue of dynamic deadlines of sensor nodes focusing on minimizing number of buffer overflows by concentrating either on overflow times or cost or proportionate mix-up of both overflow time and cost as desired (For details, refer [1]) using single mobile agent.

To summarize, all of the existing works discussed above will make use of either single or multiple mobile agents, like in VRP and its variants, for networks with periodic time windows or only one mobile agent, as discussed in [1], over entire network with dynamic deadlines.

3. Proposed Work

In this section, we propose a technique for partitioning the entire network into two partitions based on threshold value of overflow time of sensor nodes. Two mobile agents, instead of single mobile agent as suggested in any of the related works discussed in section 2, will be used to meet deadline constraints of each partition. Each mobile agent collects data in the partition in which it is deployed. Data collected by both mobile agents can be summed up at Processing Element for further processing. This, as it can be seen from results given in section 4 in later part of this paper, reduces total number of buffer overflows from both partitions than the number of buffer overflows occurring in original single network partition. Higher number of visits to sensor nodes thus achieved in each partition, put together from each partition, leads to higher amount of signal reception. Higher amount of signal reception means more accurate detection of variations in the parameters in which network is interested.

Proposed technique is given as an algorithm as outlined below for the sake of clarity of understanding.

3.1 Problem Formulation

We are given the following:

- A wireless sensor network of n nodes: $node[1..n]$
- A matrix $cost[1..n][1..n]$ that denotes the time taken to go from one node to another
- A vector that contains buffer overflow times, $Overflow_time[1..n]$. The i^{th} element of this vector determines the time at which the buffer in the i^{th} node in the network will overflow.
- *Overflow Time Threshold Value, $overflow_threshold$*
- A starting node $node_0$.

We make the following assumptions:

- The vector $overflow_time[1..n]$ consist of integer entries.
- At time $t = 0$ all the buffers of the sensor nodes start filling up.

- The actual data transfer time from the sensor node to the mobile element is negligible.

Proposed work in this paper tries to solve the problem of partitioning the network into two sub partitions each containing a subset of sensor nodes. Partitioning has to be done, based on the overflow threshold value given, such that one of the new partition of network contains set of sensor nodes with associated overflow time less than or equal to given overflow threshold and other partition contains set of sensor nodes with associated overflow time above the given overflow time threshold value. Once the partitioning is complete, use one mobile agent in each partition to collect data from sensor nodes in that network partition. Sequence of visits to sensor nodes by mobile agent in each partition can be suitably determined by applying any one of the deadline based algorithms proposed in [1].

Proposed algorithm for solving network partitioning along with mobile agent scheduling for data collection is summarized as follows:

Proposed algorithm:

Input: overflow_time[1..n], overflow_threshold,
cost[1..n][1..n]

Initialize:

Current_time = 0, current_node = start_node, j=0, k=0,
deadline1[1..n] = [0...0], deadline2[1..n] = [0...0]

BEGIN

∞/ Repeat the following for i = 1 to n

2. For every overflow_time[i]
3. If overflow_time[i] > overflow_threshold then,

--deadline2[k] = overflow_time[i]
-- k = k + 1;

Else

--deadline1[j] = overflow_time[i]
-- j = j + 1;

i = i + 1

4. Repeat the following in each partition simultaneously
Choose the node i != current_node whose deadline is closest

5. If deadline[i] < current_time + cost[current_node][i]
 – Declare failure and stop

6. Else

– current_time+ = cost[current_node][i]
– current_node = i
– deadline[i] = current_time + overflow_time[i]

END

Description:

Algorithm summarized above is modified version of original algorithm given in [1]. In Initialize part, two new vectors deadline1[1..n] & deadline2[1..n] are initialized with zeroes. After completing Step 1, the original network will be split into two partitions based on overflow time threshold mentioned. While partitioning deadline1[1..n] & deadline2[1..n] vectors get filled up

with relevant overflow time values based condition satisfied which is being checked in Step 3.

Remaining part of algorithm, i.e. Step 4 - Step 6, deals with scheduling of mobile agent visit to sensor nodes. Currently these steps are same as that of Earliest Deadline First algorithm given in [1]. Step 4 – Step 6 are expected to be executed separately in each partition simultaneously. But, these are represented as common steps with respect to both partitions. Hence, deadline vector involved in Step 4 – Step 6 has to be interpreted as deadline1[1..n] vector if network partition with sensor nodes having overflow times less than or equal to overflow_threshold is under consideration. Similarly, the interpretation of deadline[1..n] vector will be corresponding to deadline2[1..n] vector if network partition with sensor nodes having overflow times above overflow_threshold is under consideration. Also, ‘node i’ in Step 4 means that node under current consideration from each partition. Scheduling part of proposed algorithm (Step 4 – Step 6) can be changed to adopt any different deadline based mobile agent scheduling algorithm depending upon the constraints under consideration.

4. Experimental Results

Proposed technique is evaluated through C/C++ implementation & simulation. Number of buffer overflows and Number of Visits to sensor nodes by mobile agent are considered as major parameter of interest for evaluating effectiveness of proposed technique. As the parameter space is huge, we fixed some of them:

- Grid Size: It is fixed to 100x100. The speed is taken to be 1, so the cost values are same as distance between nodes. They were rounded to the nearest integer.
- Simulation time: We simulate for 1800 time units.
- Location and number of nodes n: The nodes are randomly placed on the grid. We use 10 to 100 nodes in step of 10 nodes each.
- Overflow time values : They are randomly chosen and assigned to sensor nodes.

For the purposes of evaluation, instead of stopping the algorithm when a node missed its deadline, we continue, noting this fact, and updating its deadline. Thus, the metrics used are total Number of Overflows from all sensor nodes and total Number of Visits to sensor nodes by mobile agent, for each network topology running from 10 to 100 sensor nodes, in the simulation time of 1800 time units.

The simulation results obtained are summarized as given in Tables 1, 2, & 3. In each table second column indicate the total number of buffer overflows from all sensor nodes when single mobile agent is used in given network without partitioning the network. Also, in each table third to seventh columns indicate the *sum* of total number of buffer overflows from all sensor nodes from two partitions of network, using single mobile agent in each partition, when network was partitioned based on given overflow time threshold value which is mentioned at the top of each column. Last row in each of these tables indicate the average number

of overflows for networks with 10 to 100 nodes. These average values of number of overflows are used in the graph given in Fig.1.

Topology Size	Original No. of overflows	Overflow Time Threshold				
		7	12	16	21	26
10	1060	9	240	240	240	190
20	4296	680	3574	2887	2975	2890
30	8090	1432	6455	5737	5737	5737
40	8944	736	8429	8402	8155	7908
50	12721	11898	11898	11284	10837	10864
60	14552	1942	13930	13610	13351	13695
70	19324	2600	17512	11081	17162	16870
80	20576	19607	19376	19190	19161	18482
90	24616	3871	22388	21987	21995	22057
100	25161	1534	25265	22374	24033	23692
Average	13934	4431	12907	11679	12365	12239

Table 1 Simulation Results Showing Number of Overflows for EDF Scheduling Algorithm without & with Partitioning of Network Based on Overflow Time Threshold

Topology Size	Original No. of overflows	Overflow Time Threshold				
		7	12	16	21	26
10	620	4	0	364	695	565
20	3767	608	2317	1820	1882	2076
30	6563	1275	4982	3534	3534	3534
40	8603	629	6664	6856	5991	2643
50	11539	50	10598	10347	9766	5892
60	13911	60	12398	11260	11708	7248
70	17201	70	15854	15695	14852	8747
80	19164	80	17314	17621	17657	16887
90	22198	3566	20708	21083	20217	19773
100	24287	1377	23248	21276	21800	21836
Average	12785	772	11408	10986	10810	8920

Table 2 Simulation Results Showing Number of Overflows for EDF with k-lookahead Scheduling Algorithm without & with Partitioning of Network Based on Overflow Time Threshold

Topology Size	Original No. of overflows	Overflow Time Threshold				
		7	12	16	21	26
10	1216	9	293	293	167	198
20	4188	676	3520	2855	2116	402
30	6882	1067	6304	5541	5541	5541
40	9252	730	8265	8151	7588	7588

50	11777	2929	11546	11546	10763	10763
60	14674	60	13673	13794	13448	13448
70	17750	70	17255	16973	16139	16139
80	20240	80	19274	19027	18250	18250
90	22633	90	22156	21764	21607	21607
100	25139	1457	24135	23300	23300	23300
Average	13375	717	12642	12324	11892	11724

Table 3 Simulation Results Showing Number of Overflows for MWSF Scheduling Algorithm without & with Partitioning of Network Based on Overflow Time Threshold

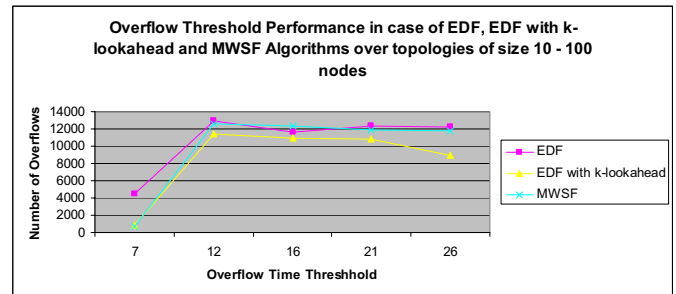


Figure 1 Graph of Average Number of Overflows v/s Overflow Time Threshold for EDF, EDF with k-lookahead and MWSF Algorithms

Graph of average number of overflows obtained for each scheduling algorithms for various values of overflow time threshold, summarized in Tables 1, 2, & 3, is given in Figure 1.

Further, Tables 4, 5, & 6 summarize the total Number of Visits to all sensor nodes obtained while simulating EDF, EDF with k-lookahead & MWSF algorithms. (Interpretation of column values is same as explained in case of earlier tables except the parameter measured). The trend of variation in total Number of Visits to all sensor nodes against the overflow time threshold is given in Fig.2.

Topology Size	Original No. of Visits	Overflow Time Threshold				
		7	12	16	21	26
10	65	371	115	115	115	124
20	62	127	117	142	126	124
30	52	116	106	119	119	119
40	71	115	128	104	110	114
50	54	105	105	130	138	120
60	69	105	126	117	126	97
70	39	119	102	97	113	116
80	61	114	111	118	116	130
90	39	101	124	128	117	100
100	61	99	95	137	130	137
Average	57	137	113	121	121	118

Table 4 Simulation Results Showing Number of Visits to all Sensor Nodes for EDF Algorithm without & with Partitioning of Network Based on Overflow Time Threshold

Topology Size	Original No. of Visits	Overflow Time Threshold				
		7	12	16	21	26
10	124	400	318	294	256	220

20	97	152	303	113	195	206
30	84	196	174	233	243	243
40	88	154	216	202	226	206
50	96	220	220	204	192	191
60	95	273	193	228	205	192
70	111	193	192	178	209	211
80	101	191	213	217	192	204
90	93	212	214	183	199	205
100	93	181	184	203	203	203
Average	98	217	223	206	212	208

Table 5 Simulation Results Showing Number of Visits for all Sensor for EDF with k-lookahead Scheduling Algorithm without & with Partitioning of Network Based on Overflow Time Threshold

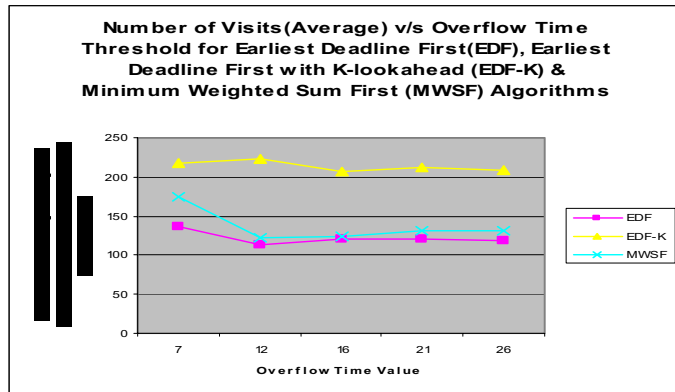


Figure 2 Graph of Average Number of Visits v/s Overflow Time Threshold for EDF, EDF with k-lookahead and MWSF Algorithms

Topology Size	Overflow Time Threshold					
	Original No. of Visits	7	12	16	21	26
10	59	364	114	114	142	142
20	62	114	112	129	134	134
30	62	364	104	116	116	116
40	68	113	139	121	137	137
50	65	112	112	102	118	118
60	66	131	132	124	125	125
70	59	130	114	129	133	133
80	64	143	123	137	138	135
90	66	137	138	134	133	133
100	65	128	128	143	143	143
Average	64	174	122	125	132	132

Table 6 Simulation Results Showing Number of Visit to all Sensor Nodes for MWSF Scheduling Algorithm without & with Partitioning of Network Based on Overflow Time Threshold

5. Discussion of Results

Graph in Fig. 1. shows that number of overflows increase rapidly as soon as overflow time threshold increases beyond value 7. Number of overflows does not decrease any more beyond this. Increase in overflow time threshold value will not have any impact on number of overflows. This implies that frequency of mobile agent service to sensor nodes can be increased by keeping overflow time threshold value in the range 7 to 12 resulting in lesser number of overflows. This observation can be

easily cross verified by looking at the number of overflows occurred in unpartitioned network with same number of sensor nodes but using single mobile agent for any given topology size.

Also, lesser number of overflows implies increase in Number of Visits to sensor nodes in the network when network was partitioned based on overflow time threshold. From Tables 4, 5, & 6 it can be verified that this implication is also true in almost all cases. Graph given in Fig. 2 shows that value of total Number of Visits decreases gradually as overflow time threshold increases from 7 to 12 and remains almost constant after that.

To summarize, as it can be observed from above results, the total Number of Overflows have always been lesser than that of what were achieved using single mobile agent over an entire network. Similarly, the total Number of Visits has always been higher than that of what were achieved using a single mobile agent over an entire network. Although the best value of overflow time threshold can not be fixed directly, suitable value of the same can be determined by running required deadline based algorithm with different value of overflow time threshold for a given topology. Disadvantage that only seems to happen is number of mobile agents we are using i.e., in terms of cost for each of these mobile agents. But compared to reduction in number of buffer overflows achieved the additional cost incurring in terms of use of more than one mobile agent is affordable. Only case where applying multiple agents can be of no use is when sensor nodes whose overflow rates are within threshold but sparsely distributed. This aspect is mentioned in below section.

6. Conclusion & Future Work

Approach of using mobile agent whose movement can be controlled is really useful for data collection from sensor nodes sampling at different rates. This paper discussed the various infeasibilities, primarily higher number of buffer overflows leading to less accurate data sensing, involved in earlier proposed approaches of using single mobile agent over entire network. In this context this paper introduced and demonstrated, in an informal way, a technique of partitioning sensor networks into two partitions based on overflow time threshold and using two mobile agents for data collection instead of single agent. The simulation results provided in this paper showed that the proposed technique will promise, almost always, good results in terms of lesser number of buffer overflows and higher of number of visits than what could be achieved using a single mobile agent over an entire network which directly helps in more accurate data sensing. There are many directions in which this work may be pursued further. First would be to define the problem in formal way to give more systematic problem formulation and proving that given solution provides increased benefit over normal approaches of using single mobile agent for entire network (in terms of mathematical proof etc). Second direction

could be exploring various other aspects for finding better way of partitioning network with different complex constraints. Another direction for research could be finding optimal number of mobile agents to be deployed in a given network.

References

- [1] Arun A Somasundara, Aditya Ramamoorthy, Mani B Srivastava. **Mobile Element Scheduling for efficient Data Collection in Wireless Sensor Networks with Dynamic Deadlines.** In Proceedings of the 25th IEEE International Real-Time Systems Symposium (RTSS 2004),2007
- [2] A. Chakrabarti, A. Sabharwal, and B. Aazhang. Using predictable observer mobility for power efficient design of sensor networks. In *The second International Workshop on Information Processing in Sensor Networks (IPSN)*, 2003.
- [3] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2002.
- [4] A. Kansal, A. Somasundara, D. Jea, M. Srivastava, and D. Estrin. Intelligent fluid infrastructure for embedded networks. In *The Second International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2004.
- [5] R. C. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: Modeling a three-tier architecture for sparse sensor networks. In *IEEE Workshop on Sensor Network Protocols and Applications (SNPA)*, 2003.
- [6] T. Small and Z. Haas. The shared wireless infostation model a new ad hoc networking paradigm (or where there is a whale, there is a way). In *The Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2003.
- [7] W. Zhao and M. Ammar. Message ferrying: Proactive routing in highly-partitioned wireless ad hoc networks. In *The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS)*, 2003.
- [8] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *The fifth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2004.
- [9] P. Toth and D. Vigo, editors, “*The Vehicle Routing Problem*”, Society for Industrial & Applied Mathematics (SIAM),2001.
- [10] M. Solomon, “Algorithms for the vehicle routing and scheduling problem with time window constraints”, *Operations Research*, 35(2), Mar-Apr 1985.
- [11] P. Toth and D. Vigo editors, “*The Vehicle Routing Problem*”, Society for Industrial & Applied Mathematics (SIAM), 2001.
- [12] www.jamesreserve.edu (james san jacinto mountains reserve
- [13] N. Bansal, A. Blum, S. Chawla, and A. Meyerson, “Approximation algorithms for deadline-tsp and vehicle routing with time-windows”, In *36th ACM Symposium on Theory of Computing (STOC)*, 2004.
- [14] Yaoyao Gu, Doruk Bozdag, Eylem Ekici, Fu’sun O’ zgu’ner, Chang-Gun Lee, “Partitioning Based Mobile Element Scheduling in Wireless Sensor Networks”, *IEEE International Conference on Mobile Computing*, 2005.
- [15] Zheng wei, Liu sanyang, Kou xiaoli, and Qi xiaogang, “Mobile agent routing algorithm in dynamic sensor network”, *IEEE International Conference on Communications and Mobile Computing*, 2010.
- [16] Ryo Sugihar, and Rajesh K. Gupta, “Optimal Speed Control of Mobile Node for Data Collection in Sensor Networks”, *IEEE Transactions On Mobile Computing*, Vol. 9, No. 1, January 2010