A Smart Data Forwarding Method Based on Adaptive Levels in Wireless Sensor Networks

Dang Hai Hoang, Thorsten Strufe*, Thieu Nga Pham**, Hong Ngoc Hoang*,
Chung Tien Nguyen***, Van Tho Tran**, Immanuel Schweizer*
Ministry of Information and Communications, Hanoi, Vietnam
Email: hdhai@mic.gov.vn
*TU Darmstadt, FB Informatik, Germany
**Hanoi University of Civil Engineering, Information Technology Faculty, Vietnam
***Academy of Cryptography Techniques, Hanoi, Vietnam

***Abstract*—This paper presents a smart data forwarding method based on adaptive levels in order to collect data in a wide area with a limited number of sensors in wireless sensor networks (WSNs). WSN nodes move on predefined trajectories. In comparison to other works, each WSN node is assigned an adaptive level, which is frequently updated based on levels and weights of other neighbor nodes. Measured data will be forwarded from nodes with higher levels on the outermost trajectories to nodes with lower levels on inner trajectories, until they reach the center. The proposed method has been tested with eight sensor nodes and one base station to cover an area of 14.6 km^2 of an urban district of Hanoi City.**

***Index Terms*—Data forwarding, data collection, adaptive level and weighted method, communication protocol.**

I. INTRODUCTION

Wireless Sensor networks (WSN) have gained prominence in recent years. For environmental monitoring applications, e.g. traffic-generated pollution data monitoring, WSN nodes can collect measurements from the roads, communicate with other WSN nodes in order to forward measured data to the monitoring center. The communication can be either one-hop from a WSN node to a sink node, i.e. the base station, or multi-hop using one-hop communication over several WSN nodes [1]. Theoretically, WSN nodes using IEEE 802.15.4 standard can send data over a possible distance of up to 700 meter in line-of-sight [2]. However, in practical deployments the range of ZigBee is usually less than a few hundred meters [3][4] due to environmental conditions. Traditional WSN architectures usually use a significant number of WSN nodes to cover a wide area of the monitoring environment, but that is not effective [5]. Recently, mobility is introduced, since it is useful for sparse WSNs, in which special Mobile Elements (MEs) are usually used to gather data from ordinary sensor nodes. Similar approaches are in the context of opportunistic networks (see survey in [6][7]).

However, mobility brings significant challenges, such as contact detection, etc. in sparse WSNs [6]. That is why, almost proposals for sparse WSNs have focused on the concept of static sensor nodes and MEs. MEs are mobile elements such as Mobile Data Collector (MDCs), or Data MULEs, or Mobile Relays (MRs) (see [6][7]), which are installed in buses, trams, or cabs. They have the task to carry data from static sensor nodes to the base station. Obviously, these approaches use a number of fixed WSN nodes. Due to the limited radio range, the number of deployed WSN nodes can be very large. On the other hand, MEs have to move along predefined routes to collect data from fixed WSN nodes. If buses or trams do not operate on these routes, these approaches are less effective. A reasonable approach for data collection in sparse WSNs requires that MEs should communicate with each other to forward data to the center. Another approach uses the concept of mobile peers, which are (unlike MEs) ordinary mobile sensor nodes in WSNs. Mobile peers have been used in ZebraNet for wildlife tracking (see [6]), where MEs move on random routes. These approaches still received little attention in the literature.

A question that remains open in practice is how to collect environmental data in a wide area, with only a limited number of WSN nodes. In this paper, we propose a smart data forwarding protocol to solve the above problem in sparse WSNs. Like other approaches using MEs, we divide the monitoring area into various trajectories. However, our design is different from other approaches with MEs (e.g. [1][3-8]). Firstly, there are no static sensor nodes. Second, our MEs are assigned by adaptive levels and weights, which are frequently updated, whenever two nodes meet (i.e. levels and weights are increased or decreased depending on connectivity condition and forwarding ability). Data forwarding occurs only from nodes with higher levels to nodes with lower levels. Unlike mobile peers, our WSN nodes move on these predefined routes. This paper is organized as follows. Section 2 presents motivation and related works. Section 3 outlines the protocol design. Section 4 describes a case study deployment. Finally, Section 5 concludes the paper.

II. MOTIVATION AND RELATED WORKS

A practical solution for pollution data monitoring in a big city requires several issues to be solved. Using the concept of only MEs on fixed routes, we define the trajectories based on the streets with most traffic pollution. For the protocol design, we have to investigate further issues including: level granting, adaptive changing of levels based on neighbor discovery,

978-1-4799-0540-9/13/$31.00 ©2013 IEEE

solution for the issue of equal levels, determining the next destination, synchronization between WSN nodes, etc. These issues have been investigated in other works with more and less differences. In following, we give an overview of the most relevant related works (see [6-8] for reference).

MDCs are mobile nodes, which move on predefined routes to collect data generated from static sensor nodes. Different types of MDCs were indicated in [6-8], which need a number of static sensors to collect data. Modifications are Mobile Sinks (MSs) and Mobile Relays (MRs). Data Mules (or Mules) are MRs that carry data from static nodes to the access point and may be a part of the network infrastructure. Similar approaches of opportunistic networks were indicated in [6]. ZebraNet was a monitoring network for wildlife, which allows tracking animal positions. The concept using mobile peers is comparable to our proposal. However, unlike ZebraNet, our MEs move on predefined routes. Almost approaches use the predefined forward direction from static nodes to MEs. In contrast, our MEs can be assigned adaptive levels as explained in Section I. Thus, the forward direction can be changed. In our best knowledge, there is still no work addressing this matter.

Some approaches proposed 3G networks for sensing with mobile phones. However, they have higher cost and are not effective in comparison to other short-range WSNs in our case.

III. Protocol Design

A. Overview of the Protocol

The monitoring area is divided by N trajectories depending on the need of monitoring data at the specific area, e.g. high traffic density of the roads. WSN nodes move on predefined trajectories. For the design, at least one trajectory should be in the coverage of the base station. Moreover, any two trajectories should have at least one overlapping road distance, so that WSN nodes are enable to transmit data to each other.

The main principles of our protocol are as follows: a) each WSN node have an adaptive level and a weight; b) data is only sent from a WSN node with higher level to the other one with lower level; c) if a WSN node sees many WSN nodes having the same level, it then sends data to the node that has higher weight; d) the data forwarding process repeats until the measured data reach the base station, which is assigned level zero. For protocol design, we should consider the following issues: How to assign levels to nodes, when they joins a trajectory? How to change the levels? Which neighbor will be selected for data forwarding? What happens, if a node received data from other nodes with higher levels, but it cannot forward data to the inner trajectory due to some reasons? We shortly describe our solution to these issues as follows.

Adaptive level assignment: Each sensor node will be assigned a level, not the trajectory. Adaptive level means that it can be changed in following cases. First, since sensor nodes are intended to forward data to the center, their levels will always be improved in order to be center-oriented. In principle, node tries to minimize its level. Second, each node changes only its level, when it can check with any other node in its coverage. Initially, all nodes set their level to be minus one ($L=-1$). This level is recognized as invalid. If a node sees the base station, its level is set to one. If a node meets another sensor node with an already assigned level ($L \neq -1$), this node will have the same level of the other node plus one. If a node meets more other sensor nodes, its level will be equal to the minimum level it observed plus one. If a node meets two or more nodes with identical levels and this level is the minimum level it observes, it will chose the node with the higher weight as next hop. The weight for each node is initiated to K, where K is a fixed constant, and will be updated as follows. After each successful data transfer in the sending phase, the weight of corresponding WSN node will be increased by K. If this node cannot send, but only receives data from other nodes it observes, its weight is decreased by one. If this node can only receive, its weight will be continuously decreased. After a certain time, its weight will get negative. In this case, the level of corresponding node will be set to minus one as same as in the initial stage and the WSN node should reassign its level again. In order to avoid the quick decrement of weight to a negative value, and consequently the re-initiation of a node, we should choose an enough large K for postponing the decrement process. This value can be chosen based on the number of intermediate nodes, which a WSN node can meet, as well as the frequency of meeting.

Due to nodes failing, or connectivity problem, a node might lose its connection to the next hop, i.e., a node with lower level of the inner trajectory. It might still receive data from other nodes with higher levels, but cannot forward it any longer. This node must find a new route in order to forward data to the inner trajectory. In this case, it must change his level to minus one (same as initial state). Thus, his level will be updated to higher level in the next meeting with another node.

B. Protocol Description

Our protocol (called Smart Data Forwarding Protocol based on Adaptive Level - SDFPAL) has five states: Initiating, Detecting, Sending, Receiving and Measuring. Each loop of five phases will take about 100 seconds.

Initiating State: will last for a few seconds after power-on. Each node initializes its node identifier (ID) and its level by -1, its weight by K. These values will be picky-backed in the beacon signal sent to other nodes, thus there is no overhead.

Detecting State: WSN node scans the network to find possible neighbors. The scan time lasts for about 20 seconds, depending on the size of network. After detecting neighbors, WSN node will update its level as follows.

$$L_i = \begin{cases} 1 & if\ ID_j = BS\_ID \\ L_{\min} + 1 & if\ L_i = -1\ \&\ L_{\min} \geq 0\ or\ L_i \geq L_{\min} \\ -1 & if\ L_i \leq L_{\min}\ \&\ W_i < 0 \end{cases} \quad (1)$$

Where ID_j is the ID of node j belonging to n current neighbors of node i, BS_ID is the ID of the base station, L_i is the current level of node i, W_i is the current weight of node i, L_{min} is the minimum level of n neighbor nodes, that is:

$$L_{\min} = \min_{j \in n} \{L_j\} \quad (2)$$

Sending State: Data forwarding is center-oriented, i.e. nodes try to send data to inner trajectory. The next destination is either base station or any neighbor. After successful transfer of m packets, the weight of node is updated according to Eq.3.

Receiving State: WSN node tries to get data from other neighbors. After successful receiving n packets, the weight of node will be updated according to Eq.3.

$$W_i = \begin{cases} W_i + K & \text{if } m > 0 \\ W_i - 1 & \text{if } n > 0 \end{cases} \quad (3)$$

Measuring State: WSN node samples the environmental data and stores them into local memory.

C. Illustration of the Protocol using an Example

We illustrate the protocol operation using an example in Fig. 1 with one base station, three trajectories (a) or four trajectories (b). Sensor nodes S_1, S_2, S_3, ... with corresponding level L_1, L_2, L_3... on trajectories R_1, R_2, R_3...

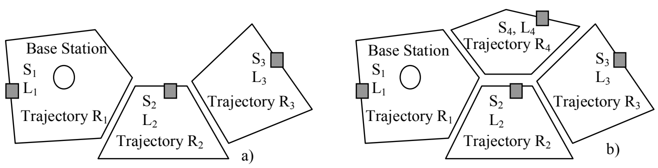

Fig. 1. Example with a) three trajectories and b) four trajectories

1) During the initial stage, each node has the level $L_i = -1$ and a weight of K. K is a constant value as describe above.
2) Whenever S_1 sees the base station, its level is set to $L_1=1$, thus, S_1 is possible to send its collected data to the base station.
3) If S_2 sees S_1, its level is set to $L_2=L_1+1=2$. S_2 is possible to send its data to S_1.
4) If S_4 sees S_2 before seeing S_1 (Fig.1b), its level is set to $L_4=L_2+1=3$. Thus, S_4 can send its data to S_2.
5) But if S_4 sees S_1 before seeing S_2, its level is set to $L_4=L_1+1=2$. In this case, S_4 and S_2 will have the same level that is equal to 2. Both S_2 and S_4 can send data to S_1.
6) Following case 4), if S_3 sees S_2 before seeing S_4, its level is set to $L_3=L_2+1=3$. Both S_3 and S_4 can send data to S_2. If S_3 sees S_4 before seeing S_2, its level is set to $L_3=L_4+1=4$. Thus, S_3 is possible to forward its data to S_4 and S_2.
7) Following case 5), assume that S_3 sees S_2 and S_4 with the same level $L_2=L_4=2$. If the weight of S_2 larger than the one of S_4, the level of S_3 is set to $L_3=L_2+1=3$. Otherwise, $L_3=L_4+1=3$. Thus, S_3 can send its data either to S_2 or S_4, respectively.
8) Assume that S_2 has $L_2=2$ and S_4 has $L_4=3$. If S_3 meets both S_2 and S_4, S_3 will choose the node with minimum level among its neighbors, i.e. S_2 in this case.
9) If S_2 received data from S_3, but could not forward to S_1 due to some problem, S_2 changes his level to $L_2= -1$. If S_2 meets S_4, his level is updated to $L_2= 3$, thus S_2 could forward data to S_4. That is, S_2 uses a new route S_2-S_4-S_1 instead of the old route S_2-S_1. S_2 updates his level to $L_2= 2$, when it sees S_1 again.

D. Protocol Performance

Denote N_{loss} the number of lost packets, $N_{received}$ the number of received packets at the destination, N_{send} the number of packets sent from a source, the packet loss rate is as follows.

$$P_{loss} = N_{loss} / (N_{loss} + N_{received}) = N_{loss} / N_{send} \quad (4)$$

Packet delay is defined as the time different of the receiving time of a packet and the sending time of the corresponding packet.

$$\text{Delay} = t_{rec} - t_{send} \quad (5)$$

Throughput θ_{i-j} of packet transfer from node ID_i to node ID_j in a time interval $(t_2 - t_1)$ can be calculated as follows.

$$\theta_{i-j} = (N_2 - N_1) / (t_2 - t_1) \quad (6)$$

Where N_2 and N_1 is the number of packets sent at time t_2 and t_1, respectively. In fact, throughput is the number of sent packets in a time interval, e.g. one second in our deployment.

IV. Case Study Deployment

Our protocol was deployed for an urban district of Hanoi City using a ZigBee mesh network based on a Meshlium and eight Waspmotes [4]. The nodes move on eight trajectories in order to cover an area of about 14.6 km^2 as shown in the Fig. 2. The trajectories have been designed so that each mote can meet at least one other mote with enough time to be able to send the data. Additionally, two of the trajectories have some parts, on which Waspmotes can meet the Meshlium. Each mote runs on a trajectory and is numbered according to the trajectory number, i.e. from 1 to 8. The moving speed of each node is about 10 km/h. We consider two requirements as follows:

(1) Test of protocol operations in various circumstances;

(2) Test of real-time forwarding, i.e. the base station should receive new data after every hour.

Fig. 2. Implementation of SDFPAL with eight trajectories

A. Test of Protocol Operations in various Circumstances

For the requirement (1), we built two scenarios as follows.

Scenario 1: All Waspmotes move on 8 trajectories as shown in Fig. 2. We denote a node with ID (e.g. ID8 for the node S_8), the level with L (e.g. L_3 for the level 3). We investigated the forwarding route ID8(L_3)-ID3(L_2)-ID1(L_1)-Meshlium. After a time of measuring, ID8 saw ID3 on an overlapping road distance of 665m (meeting time of about 4 minutes). After that, ID3 moved alone on his trajectory that has a distance of 3066m. Then, ID3 saw ID1 on an overlapping road distance of 365m (meeting time is about 130 seconds).

Figure 3 shows the packet delay by forwarding packets from ID8 to ID3 (solid line) and from ID3 to ID1 (dotted line). The average delay varies from 1.0s to 2.0s for both routes. The calculation of delay is following Eq.5. Figure 4 presents the throughput (packets/s) of data transfer from ID3 to ID1. We calculate the throughput for each interval of one second according to Eq. 6. As presented, throughput of data transfer from ID3 to ID1 is from 0.5 to 1.5 packets/s.

In scenario 1, we have sent 121 packets from ID3 to ID1. Among them, we counted that 10 packets are lost. Thus, the corresponding packet loss rate is about 10%. In our experiments, the loss rate for data transfer from ID8 to ID1, from ID8 to ID3 is 18% and 20%, respectively. We calculated these values using Eq. 4 as presented above. This loss rate is reasonable, since the Waspmotes can have a lot of measurements at a location, thus the loss of some packets might not affect the data processing at the center.

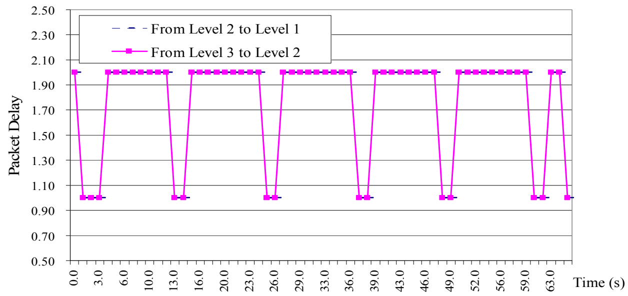

Fig. 3. Packet delay by forwarding from ID8 and/or ID3 to ID1

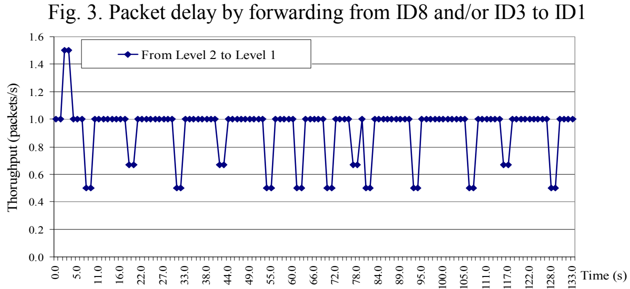

Fig. 4. Throughput of data transfer from ID3 to ID1

Scenario 2: To evaluate the adaptive level change, we have stopped the operation of Waspmote 2 and 3. The results are given in Table I.

TABLE I. RESULTS OF ADAPTIVE LEVEL CHANGING SCENARIO

Waspmote ID	Level Scenario 1	Level Scenario 2	Waspmote ID	Level Scenario 1	Level Scenario 2
1	1	1	5	2	5
2	1	Stop	6	2	3
3	2	Stop	7	3	3
4	2	2	8	3	4

Explanation: Since ID6 can only meet ID4 (level 2), its level has changed to 3 instead of 2. Since ID8 can meet ID6 and ID7 (level 3), its level has changed to 4 instead of 3 as before. Since ID5 can meet ID8 (level 4), its level has changed to 5 instead of 2 as before. That is, ID5 was not an intermediate node for ID8 anymore. Instead, it used ID8 as an intermediate node for sending its data to the Meshlium via indirect routes (5 – 8 – 6 – 4 – 1 or 5 – 8 –7 – 4 – 1).

B. Test of Real-time Forwarding

We designed the trajectories using the following principle. The total road each packet, starting at the location of collecting and ending at the base station, should be always less than 10km. Since Waspmotes move with a speed of 10km/h, we can ensure that data from the farthest Waspmotes can be forwarded to the center. Thus, the base station is ensured to receive updated data after every hour. We have tested with the farthest Waspmote (ID8) within an hour and showed that the packet loss rate was 18%, which is an acceptable rate.

V. CONCLUSION

The paper presented a smart data forwarding protocol for collecting measured data in a wide area with a limited number of WSN nodes. Instead of using fixed nodes, we use the MEs as intermediate nodes. Data is forwarded based on adaptive levels granted to each node. This enables us to cover a large monitoring area, using relatively few nodes. We addressed several issues such as assignment of levels, adaptive change of levels, decision of next destination in case of multiple intermediate nodes, failed transmission of an intermediate node and synchronization of different states of multiple nodes. The proposed method has been tested using a ZigBee network with one Meshlium base station and eight Libelium Waspmotes to cover an area of 14.6 km^2 of an urban district of Hanoi City. The results indicate that our protocol can be used to forward the data from different WSN nodes to the base station.

ACKNOWLEDGMENT

This research project has been supported by the Alexander von Humboldt Foundation and co-funded by the DFG as part of the CRC 1053 MAKI. We would like to thank the sponsors for offering the opportunity, by which German and Vietnamese researchers can cooperate in this joint project.

REFERENCES

[1] I. Schweizer, M. Wagner, D. Bradler, M. Muehlhaeuser, T. Strufe, "kTC - Robust and Adaptive Wireless Ad-hoc Topology Control", IEEE International Conference on Computer Communication Networks (ICCCN), 2012.

[2] http://www.libelium.com/development/waspmote/documentation/waspmote-datasheet/.

[3] A.Hardisty, B. Goossnes, L.N. Ambu, C.Gwilliams, A. Preece, "K-Has: an architecture for using local and global knowledge in wireless sensor networks", The International Conference on Informatics and Applications (ICIA2012), pp.331-341.

[4] D. Suciu, "A study of RF link and coverage in ZigBee", Scientific Bulletin of the "Petru Major" University of Targu Mureq, Vol. 7 (XXIV), No. 1, 2010, ISSN 1841-9267, pp.5-10.

[5] M. Keller, M. Woehrle, R. Lim, J. Beutel, L. Thiele, "Comparative performance analysis of the PermaDozer protocol in diverse deployments", Proc. of the 6th IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp 2011), pp.969-977.

[6] M.Francesco, S.Das, "Data Collection in Wireless Sensor Networks with Mobile Elements: A Survey", Journal ACM Trans. on Sensor Networks, Vol.8, Iss. 1, Aug.2011, NY, USA.

[7] K. Shah, M. Francesco, G. Anastasi, M. Kumar, "A Framework for Resource-Aware Data Accumulation in Sparse Wireless Sensor Networks", Computer Communications, Vol.34, Issue 17, Nov. 2011, pp. 2094–2103.

[8] http://www.opensense.ethz.ch/trac/.

[9] K.Kondepu, G.Anastasi, M.Conti, "Dual-Beacon Mobile-Node Discovery in Sparse Wireless Sensor Networks", IEEE Symp. on Computers and Comm. (ISCC), 28.6-01.7, 2011, p.796-801.