A DTN-Based Sensor Data Gathering for Agricultural Applications

Hideya Ochiai, *Member, IEEE*, Hiroki Ishizuka, Yuya Kawakami, and Hiroshi Esaki, *Member, IEEE*

(Invited Paper)

***Abstract*—This paper presents our field experience in data collection from remote sensors. By letting tractors, farmers, and sensors have short-range radio communication devices with delay-disruption tolerant networking (DTN), we can collect data from those sensors to our central database. Although, several implementations have been made with cellular phones or mesh networks in the past, DTN-based systems for such applications are still under explored. The main objective of this paper is to present our practical implementation and experiences in DTN-based data collection from remote sensors. The software, which we have developed for this research, has about 50 kbyte footprint, which is much smaller than any other DTN implementation. We carried out an experiment with 39 DTN nodes at the University of Tokyo assuming an agricultural scenario. They achieved 99.8% success rate for data gathering with moderate latency, showing sufficient usefulness in data granularity.**

***Index Terms*—Delay-disruption tolerant networking (DTN), experiment, sensor data gathering, sensor networks.**

I. Introduction

AGRICULTURAL researchers and farmers deploy sensors at their remote agricultural-fields to obtain the data of temperature, humidity, soil moisture and so on. Automatic collection of those data greatly helps their analytical works. Currently, they are relying on network providers, for example, cellular phone network to achieve such automatic collection from their remote sites, which is not feasible for most of the farmers due to operational cost.

We have explored a delay-disruption tolerant networking (DTN)-based system that collects sensor data from remote sites without relying on network providers. DTN, which stands for delay (or disruption) tolerant network, was originally proposed for interplanetary communication [4]. It is now widely acknowledged as a framework that can be applied to village-to-village communications [7], vehicular communications ([10], [13]), and sensor networks ([12], [22]). Focusing on the application to sensor networks, we contribute to the research community by showing our implementation-based experiment.

Several approaches exist for collecting data from such remote sensors. However, we must keep in mind that sensors should cover the agricultural fields sparsely but the number would become large because the field itself is large. We must collect data from a number of sparsely deployed sensors. Thus, attaching a cellular phone device or satellite communication device to every sensor is not feasible for normal farmers because of the communication fee. Setting up wireless mesh network with ad-hoc technologies (e.g., optimized link state routing protocol (OLSR) [6] or ad hoc on-demand distance vector (AODV) [16]) could be used but it requires a huge number of relay nodes in the large area (i.e., we must densely deploy them) so as to extend network connectivity to sensors. The DTN approach, which uses the mobility of tractors and farmers, allows the collection of data from those sparsely deployed sensors.

We assume that sensors, vehicles (e.g., tractors, farmers) and the homes of farmers have short-range radio communication devices. Farmers and researchers use their tractor or their foot to visit their remote sites constantly: e.g., every day or every week. The radio device on such vehicles contacts with sensors at the remote sites, and returns to their home position. By making use of the ad hoc communication chances, they collect remote sensor data to their data server.

This paper presents our field experience on sensor data gathering conducted in the University of Tokyo assuming such an agricultural scenario. We deployed five weather sensors sparsely in the campus, and vehicular nodes collaboratively collected data from them to our central database. Totally, we have introduced 39 DTN nodes for the experiment and they have achieved 99.8% success rate for data collection.

This paper is organized as follows. Section II describes the system overview for DTN-based data gathering from agricultural-fields. Section III presents the routing algorithms and management mechanism for delivering data over the network. Section IV shows the experiment and its result. Section V provides discussion and related works. Finally, we conclude this paper in Section VI.

II. Data Gathering With DTN From Remote Sensors

Fig. 1 shows the overview of the data collection mechanism from remote agricultural-fields. Here, we assume that this system is operated by a research laboratory or a farmer. There is a data server at their home position that archives the data

Manuscript received January 31, 2011; revised September 21, 2011; accepted September 23, 2011. Date of publication October 03, 2011; date of current version October 26, 2011. This is an expanded paper from the IEEE SENSORS 2010 Conference and was published in its proceedings. The associate editor coordinating the review of this paper and approving it for publication was Dr. Thomas Kenny.

The authors are with the University of Tokyo, Tokyo, Japan 1138656 (e-mail: jo2lxq@hongo.wide.ad.jp; isi@mcl.iis.u-tokyo.ac.jp; yuyarin@hongo.wide.ad.jp; hiroshi@wide.ad.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSEN.2011.2170562

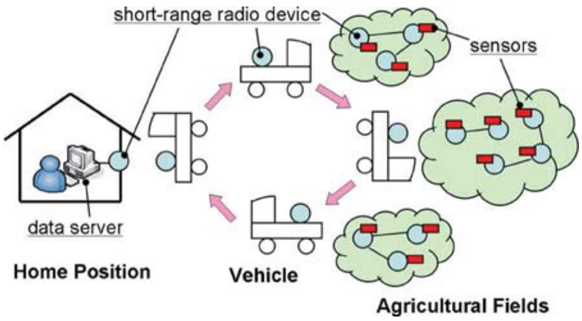

Fig. 1. By using the movement of vehicle, DTN allows data collection from remote agricultural field sensors.

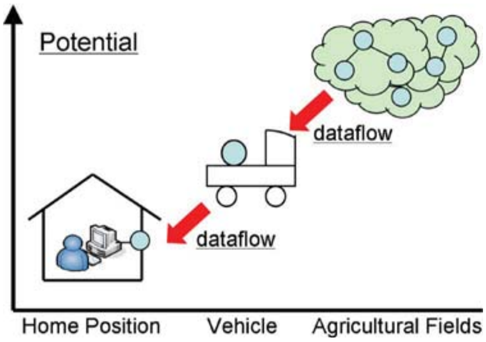

Fig. 2. By setting higher potential to remote sensors and lower potential to data server, sensor data autonomously gathers at lower potential nodes, just as water flows from higher places to lower places.

collected from their remote sites. Their tractor moves between the home position and the remote sites. Each node (i.e., data server, vehicle and sensor) has a Wi-Fi interface, and the Wi-Fi interfaces are working in "ad hoc" mode. Thus, when they come into radio communication range, they find each other and can exchange data. For example, when the tractor arrives at the remote sites, it gets the history record of the sensor data and saves into its local storage. After return, it transfers the stored data to the data server. Even though the delivery of data is not real-time (because it is physically bound to the movement of the vehicle), it is sufficient for data analysis purposes.

We apply potential-based routing (PBR) approach to enable autonomous data transfer. Fig. 2 shows how PBR allows autonomous data delivery from sensors to the data server by making use of the vehicle. In PBR, we introduce *potential*—a scalar value that represents a heuristic proximity from the destination. We define a rule that a node sends messages to the node that has the lowest potential among its neighbors. By setting higher potential to the sensors and lower potential to the server, the vehicle receives data from the sensors at the remote sites and provides data to the server when returned. In this way, sensor data gather at the data server by making use of the movement of the vehicle.

Each sensor submits observed sensor readings in a message form periodically to the network with specifying the destination to be the data server. For example, the content of a message sent by a sensor could be as follows:

```
Destination: 99
Source: 1
TTL: 81250
Timestamp: 2010-11-02T12:34:00+09:00
Temperature: 25.5
Humidity: 56.9
RainFall: 0.3
```

This message is composed of header part and body part. In the header part, it specifies that the destination is 99 (their data server) and shows that it came from 1 (sensor #1). This message still has 81250 second for delivery lifetime. In the body part, the message has timestamp and sensor readings. Though this was not the real format we used in the experiment (we used binary format in a UDP message), this kind of information was contained in the message.

The sensor itself does not need to care the storage or retransmission of the message. The network manages the delivery with taking care of storing and retransmitting by using the message header information.

III. Message Routing to Properly Gather Data

The previous example (e.g., Fig. 2) is actually a simple case; only one vehicle is assumed and it is easy to set higher potential to sensors and lower potential to the server. However, the problem becomes complex if the number of vehicles increases. For example, in our experiment setting, a vehicle is assigned to a certain region and it cannot get out from the region. Other vehicles work in different regions. They have overlapped areas, and they can exchange messages there. We must set appropriate potential values to them so as to properly direct the messages to their destinations.

We recognize that we can manually design and setup appropriate potential values for each node for the given deployment or movement pattern. However, we here assume that they determine such patterns in ad hoc manner and the patterns change day-by-day. Configuration of potential values to, for example, 30 nodes every day becomes a hard work. Thus, the nodes must develop proper potential values autonomously. If we could allow autonomous development of potential values at each node, the network can be applied to such a practical scenario.

Inheriting the concept of potential-based routing, we generally proposed potential-based entropy adaptive routing (PEAR) in our previous literature [14]. Thus, our field study is one of the applications of PEAR. In this section, we introduce the overview of PEAR as a data routing framework for sensor data gathering. However, because the main focus of this paper is not on the algorithms of PEAR but on the field experiences on sensor data gathering, please refer to the previous paper for more detail.

A. Potential-Based Routing (PBR)

As we described, we applied potential-based routing (PBR) to sensor data gathering. We here provide the formal definition of PBR.

Let N be a set of nodes and $nbr(n)$ be a set of neighbors of node $n(\in N)$.[1] A node has potential values, each of which is associated to a particular destination. We denote a potential

[1]In our definition, $nbr(n)$ includes n itself: i.e., $\{n\} \subseteq nbr(n)$.

value by $V^d(n,t)$: the potential of node $n(\in N)$ associated to destination $d(\in N)$ at timeslot t. $F^d_k(n)$ is the force on data from node n toward neighbor k, which we define as

$$F^d_k(n,t) = V^d(n,t) - V^d(k,t). \quad (1)$$

Then, PBR forwards data according to the following rule.
if $\max_{k\in nbr(n)}\{F^d_k(n,t)\} > \alpha$,

$$nexthop^d(n,t) = \left\{k | k \in nbr(n) \wedge F^d_k(n,t) = \max_{j\in nbr(n)}\left\{F^d_j(n,t)\right\}\right\} \quad (2)$$

else

$$nexthop^d(n,t) = \{\} \quad (3)$$

Here, α, a positive constant parameter, is the threshold of data transfer. If $F^d_k(n)$ exceeds α, the nexthop is such neighbor k that gives maximum $F^d_k(n,t)$. If not, no nexthop is provided for destination d, which means that the data should be stored in the local buffer.

B. Potential-Field Construction

In order to develop potential values autonomously, we have designed the following potential-field construction algorithm in PEAR

$$V^d(n,t+1) = V^d(n,t) + D \min_{k\in nbr(n)}\left\{V^d(k,t) - V^d(n,t)\right\} + \rho \quad (4)$$

$$V^d(d,t) = 0 \quad (5)$$

$$0 < \rho < D, \quad 0 < D < 1. \quad (6)$$

The potential of destination is always tied to 0 [(5)]. Other potential values dynamically change depending on node-contact patterns. A potential normally grows by ρ at every timeslot, but decreases when the node has encountered such node that gives lower potential [(4)]. As a whole, the destination has the lowest potential, the nodes near the destination have lower potential, and the distant nodes have higher potential. In this way, the potential values autonomously develop as Fig. 2.

D is a diffusion parameter that defines how rapidly potential changes propagate in the network. If it becomes larger, potential values decrease faster when the node has encountered lower nodes, and dissemination of low-potential information becomes faster.

C. Acyclic Message Propagation

We originally considered that forwarding data in the network was the correct delivering scheme. Data forwarding means that a node removes data from its local buffer when forwarded to the next node. In this data propagation scheme, the data possibly return to the node which it has visited before. This is widely known as *loop*, which only consumes network resources such as radio transmissions, CPU, and data buffers.

Fig. 3. DTN nodes (UTMesh) for the experiment. We configured five nodes for sensors and one node for database, used seven nodes as vehicular nodes and deployed 26 static nodes (Fig. 5 illustrates the setting of the experiment).

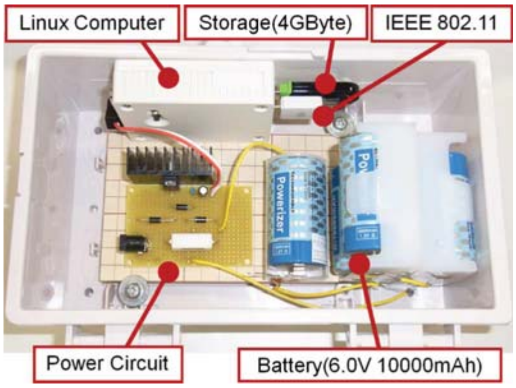

Fig. 4. A DTN node is composed of Linux computer, storage device, Wi-Fi (IEEE802.11) interface, and rechargeable battery. With Wi-Fi in ad hoc mode, DTN nodes can exchange data with neighbor nodes.

We have introduced replica management algorithm into PEAR, however, not only for avoiding such loop. The main objective was to allow *copy-based data transfer*, which propagates data in the network not by forwarding but by copying. It requires each node to have states to indicate which data are already received. It has turned out that replica management avoids forwarding loop because once a node gets data it does not receive the same data again. Though copy-based data transfer consumes network resources especially radio transmissions and data buffers, the amount of sensor data in our application is not large and practically feasible (as our experiment demonstrates in Section IV).

Actually, copy-based data transfer is effective. Spray and Wait ([19], [20]) has shown that message replication in the network reduces delivery latency and improves delivery success rate under random way point mobility models. Though it loads on the network by radio transmissions and buffer usages, it improves the availability (i.e., fault tolerance) and latency by taking several delivery paths in parallel.

IV. Implementation and Deployment

A. Experiment Setting

We carried out an experiment with 39 DTN nodes from UTMesh[2] nodes (Fig. 3). We installed our experimental soft-

[2]UTMesh: wireless mesh networking testbed at the University of Tokyo

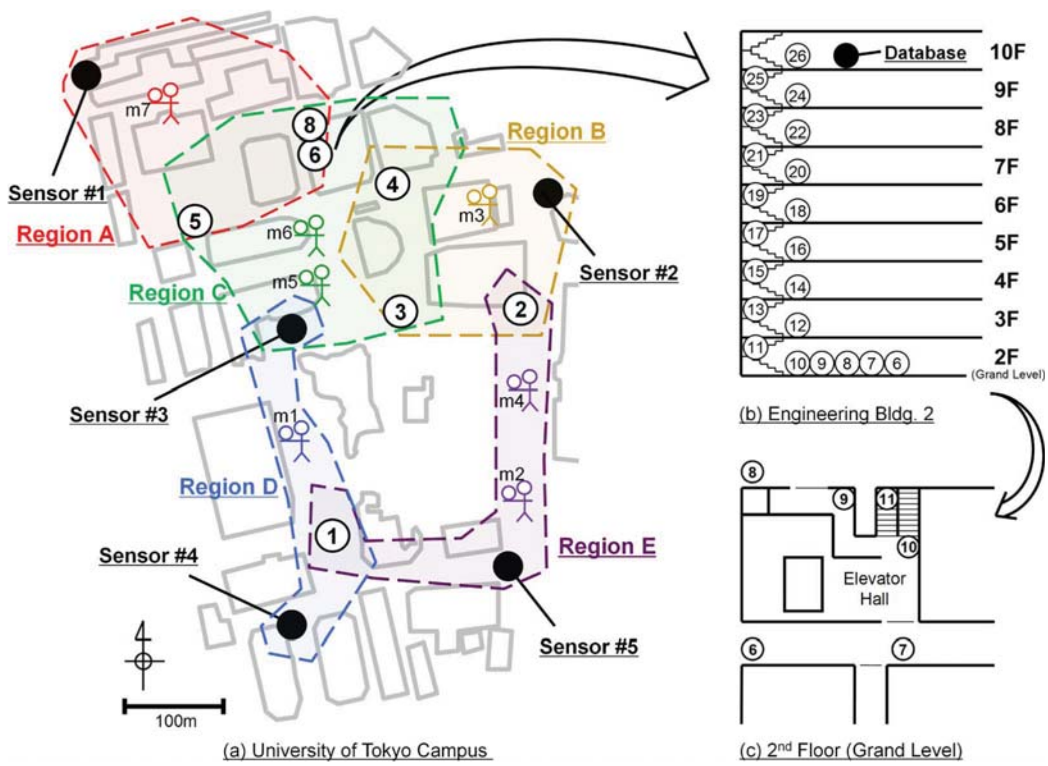

Fig. 5. The experiment configuration at the University of Tokyo. We deployed sensors (#1, . . ., #5), database, vehicular nodes $(\mathrm{m1}, \ldots, \mathrm{m7})$, and static nodes (1, . . ., 26). (a) We divided the campus into five regions, and vehicular nodes moved inside their assigned regions. We deployed several static nodes in the overlapped areas, intending that they help data propagation between the regions. (b) Database was setup at our laboratory, on the 10th floor of Engineering Building 2. We deployed static nodes along with stairs to extend the network from (c) the ground level of the building. When vehicular nodes walk around 6, 7, and 8, the static nodes absorb messages from the vehicular nodes and forward to the upper stairs.

ware for disruption tolerant networking into these nodes and deployed into the campus of the University of Tokyo. The software implemented PEAR with about 3000 lines in C source code.[3] The footprint was about 51 kbyte, which indicates that it is runnable on embedded computers. It had 3072 entries for data buffer in the experiment. We set 0.02 for ρ, 0.2 for D and 0 for α [(2) and (4)].

Fig. 4 shows the configuration of a DTN node. It has an embedded computer called Armadillo-220, which has 8 Mbyte for program memory and 32 MByte for working memory. It works with ARM9 200 MHz CPU and Linux operating system. We added an USB Wi-Fi (IEEE802.11) interface[4] for ad-hoc communication with radio-range neighbors, and an USB storage to archive the working logs. We used linux-2.6.12.3-a9-15 for its kernel image. The DTN nodes were powered by battery during the experiment.

We carried out the experiment in the following manner. As Fig. 5, we divided the campus into five regions (A, B, C, D, E). They had small overlapped areas where vehicular nodes can transfer data to the other region. We deployed five weather sensors (#1, #2, #3, #4, #5) in each region respectively and assigned vehicular nodes $(\mathrm{m1}, \ldots, \mathrm{m7})$ to the regions as the figure. We called for participation for the experiment from our laboratory students, assigned each student a region and a DTN node, asked them to be a vehicular node and to walk inside their working regions. We deployed 8 static nodes (1, . . ., 8) at the overlapped areas, intending that they help message transfer between the regions. Other static nodes (9, . . ., 26) were deployed at the stair area in Engineering Building 2 [Fig. 5(b)]. The data server was deployed at the 10th floor, where our laboratory is located.

The weather sensors used in the experiment were Vaisala WXT520. The sensor nodes observed the weather status (i.e., temperature, humidity, rainfall, wind speed and direction, air pressure) and sent them to the database every 30 s with 7200 s for delivery lifetime. Actually, each node had an unique identifier. For example, sensors were operated by $\mathrm{nodeID} = (\#011, \ldots, \#015)$. Vehicular nodes $(\mathrm{m1}, \ldots, \mathrm{m7})$ were operated by $\mathrm{nodeID} = (\#001, \ldots, \#007)$. The node ID of the data server was #099. Data transmission and routing was based on those identifiers; i.e., sensors have sent data from #011, #012, #013, #014, #015 to #099. The naming rule above such as sensor #1, sensor #2 and m1, which were different from nodeIDs, was defined just for readability.

The experiment was conducted for one day. But, we have setup 80 min for core experiment time (13:50–15:10). We first setup static nodes and sensor nodes at the specified sites, then vehicular nodes walked inside their own regions at the core time. We studied the summarized network topology, potential-field construction, delivery pattern, delivery success rate, delivery latency, and the collected sensor readings.

B. Network Topology

Fig. 6 illustrates the summarized network topology made during the experiment duration. The boldness of links indicates the summary of contact-duration between the two nodes. The vehicular nodes relayed data between static nodes and sensor

[3] http://sourceforge.net/projects/pear/files/.

[4] GW-USMicroN, Planex Communications, Inc.

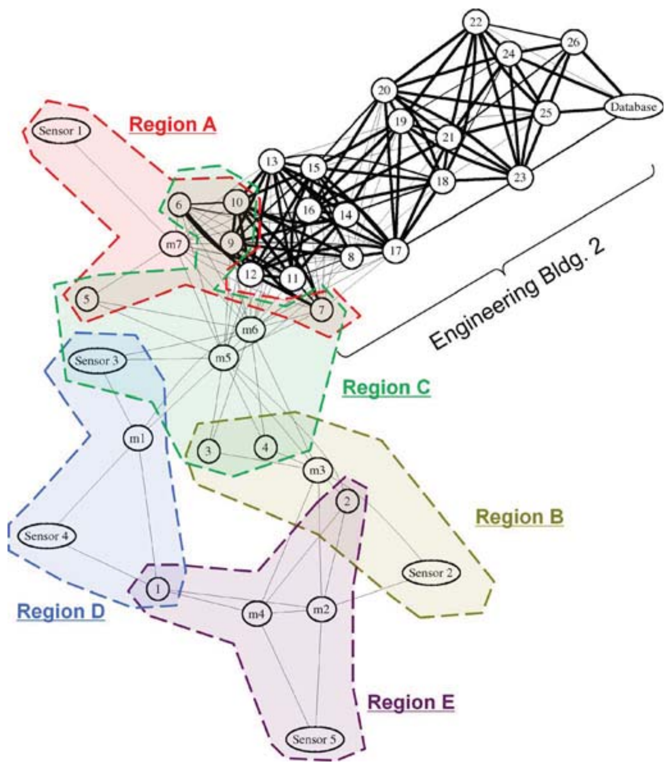

Fig. 6. The summarized network topology made in the experiment. The boldness of links indicates the summary of contact-time duration between the two nodes. The vehicular nodes relayed static nodes and sensor nodes in their regions. As a whole, all the nodes were connected into a single network.

nodes in their regions. As a whole, all the nodes were connected into a single network. The data transmitted from sensors were copied via these links (when available) and finally arrived at the database.

The static nodes deployed in the building tightly connected with each other. Vehicular nodes (m5, m6, m7) sometimes visited the entrance of the building (around static nodes 6, 7, 8). m7 visited sensor #1, m5 and m6 visited sensor #3 and static node (3 and 4). In the same way, m1 bridged data from sensor #4 to sensor #3 or static 1 (in this case, sensor #3 was working as a relay node for sensor #4).

C. Potential-Field Construction

Fig. 7 illustrates how potential values have changed during the experiment. This diagram shows the potentials of sensor nodes (#3, #4), vehicular nodes (m1, m5, m6) and static nodes (1, 6, 20). At 13:50, vehicular nodes departed from the entrance of the building to their working regions. The potential values started to increase after the departure. In the transitive phase, the vehicular nodes contacted sensor #3, static 1, and sensor #4, which had been working alone. The potentials of those nodes were high because they had not contacted any nodes in the previous several hours, but after the contacts those potentials decreased. After vehicular nodes walked inside their regions many times, the network went into a nontransitive phase. The potential values change dynamically according to the movement of vehicles. However, this diagram tells that, in the long run, the closer nodes to the destination had lower potential values (static 20 and 6) and the distant nodes had higher potential values (sensor #4, static 1, and m1).

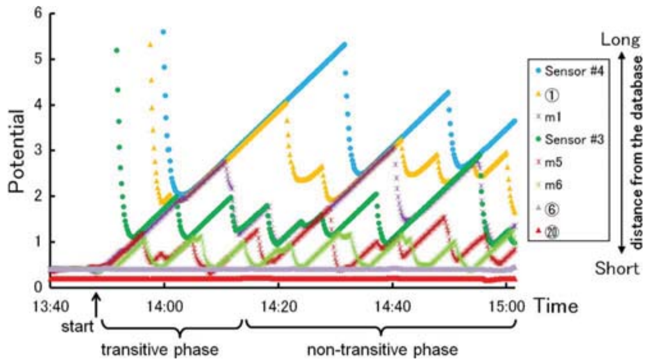

Fig. 7. Dynamics of the potential-field construction. After a while (in non-transitive phase), the closer nodes to the destination had lower potential values and the distant nodes had higher potential values.

D. Delivery Pattern, Success Rate, and Latency

Fig. 8 illustrates the data delivery pattern in the network from the sensors #1, #2, #3, #4, #5 to the database. Each arrow shows data transmissions between the nodes. The boldness indicates how many data were sent at the same time (bolder arrows sent larger number of data elements). The color identifies the source of the data. Each color corresponds as follows: $(\text{red}, \text{yellow}, \text{green}, \text{blue}, \text{violet}) = \text{sensor}(\#1, \#2, \#3, \#4, \#5)$.

This diagram tells the following dataflow.

- Vehicles took data from sensors and sent mostly to relay nodes (including sensor #3). For example, vehicle m7 took data from sensor #1 around 14:38 and sent to static 6 and 7 around 14:44. Vehicle m1 took data from sensor #4 around 14:35 and copied them to sensor #3 (relay node) around 14:41.
- Static nodes deployed in the field relayed data between the regions. For example, static 4 received data from sensor #2 and #5 via vehicle m3 (Region B) around 14:38 and copied them to vehicle m6 (Region C) around 14:43.
- Data were sometimes copied between vehicles directly: e.g., from m3 to m5, from m7 to m5, and from m5 to m6.
- Static nodes deployed in the building (static 6, ..., 26) relayed data to the upper stairs.
- Finally, the data reached the database.

Fig. 9 shows delivery success rate and delivery latency from each sensor to the database. This analysis was made in the following manner. First, we focused on the data sent between 13:30 and 14:30, then calculated the latency for each data reached at the database before 15:10. The experiment ended at 15:10 when they remained still in their own region. We got this diagram by summarizing them into delivery success rate. If some data are missing, the success rate does not reach 1.

Data were collected with moderate latency about 10–75 min, effectively using the movement of vehicles—these distributions depended on the physical movement. Finally, almost all the data were gathered at the database. The network achieved 100% delivery from sensor #2, #3, #4, #5. The delivery rate from sensor #1 was 99.2%. The vehicle who took a role in Region A did not visit sensor #1 until 14:37 though the experiment begun at 13:50. This failure of the movement probably missed a small piece of data from sensor #1. Anyway, totally, the network has achieved 99.8% delivery success rate.

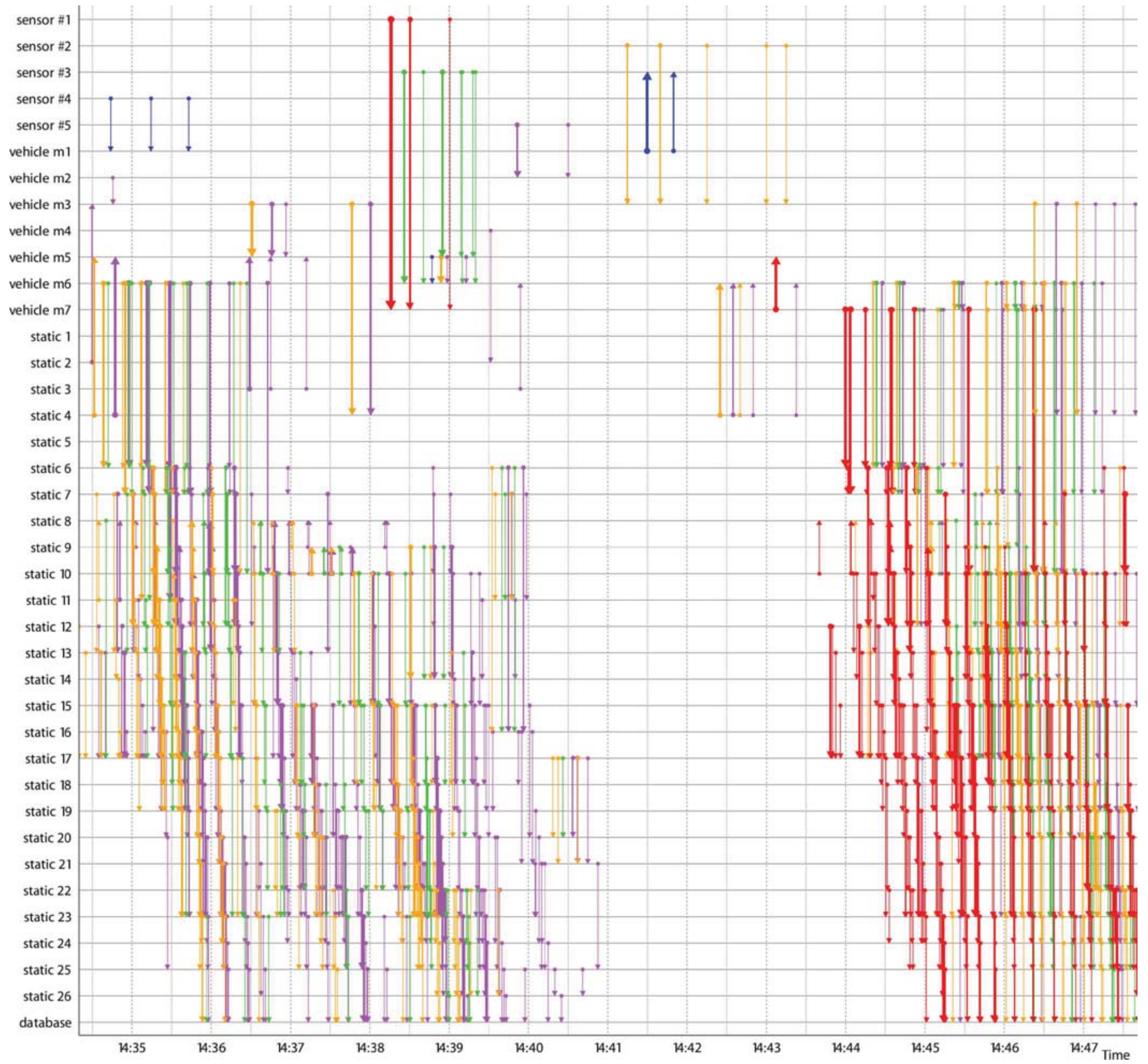

Fig. 8. Dataflow from the sensors (#1, #2, #3, #4, #5) to the database. An arrow shows data transmission between the nodes. Bolder arrows sent larger number of data at the same time. The color corresponds to the source of the data flows as $(\#1, \#2, \#3, \#4, \#5) = (\mathrm{red}, \mathrm{yellow}, \mathrm{green}, \mathrm{blue}, \mathrm{violet})$.

E. Application

Fig. 10 shows the sequence of the collected temperature. We can read that the network has collected data with sufficient time-granularity at least for agricultural use cases. Actually, there were almost no loss (see Fig. 9). Almost all the data were gathered at the database by effectively using the movement of vehicles.

According to the deployed location, the temperature patterns were quite different. The temperature of sensor #2, which was deployed under the sunlight, rapidly increased before 13:55 and decreased after that (because the sunlight was blocked off by clouds). The result indicates that the interval of readings and delivery success rate were sufficient for tracking this kind of changes.

V. Discussion and Related Work

We have demonstrated with real implementation that the application of DTN allows sensor data gathering with sufficient success rate. We organized the network for data collection from remote sensors without buying PHS or cellular phones. Instead, we used mobile nodes assuming that farmers and tractors move around in their agricultural-fields. In this work, we have studied the performance of PEAR—a data routing scheme for such network. It efficiently used the pattern of mobile nodes and achieved 99.8% success rate. The footprint of PEAR was 51 kbyte, thus, we could implement them into an embedded computer.

Sensor data gathering with the application of DTN was identified, for example, by DFT-MSN [22] and vehicular sensor networks [12]. Ren *et al.* [17] proposed a hybrid approach with

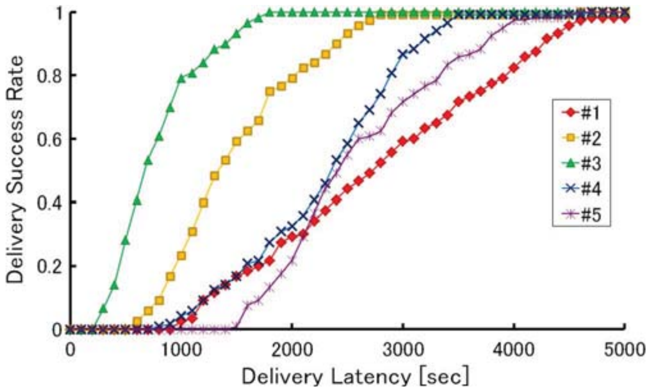

Fig. 9. Delivery rate and delivery latency from each sensor to the database. Finally, the network achieved 100% delivery from all the sensors except #1. The delivery rate from sensor #1 was 99.2%. The pattern of the latency actually depends on the physical movement of vehicles.

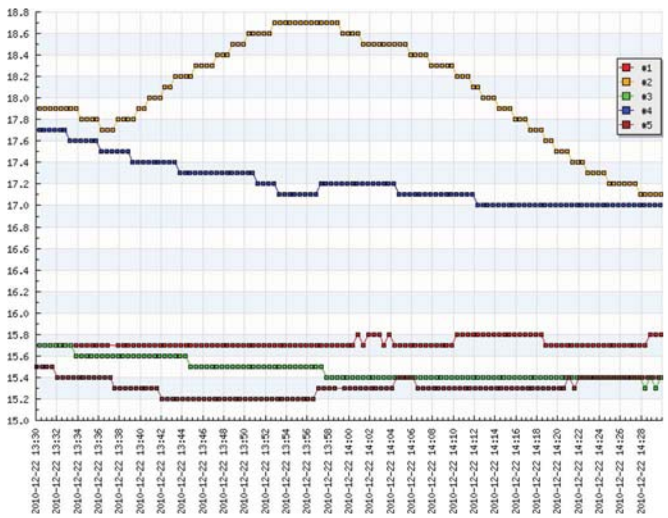

Fig. 10. Collected temperature from sensor #1, ..., #5. The experiment achieved almost 100% delivery (see Fig. 9) and provided sufficient time-granularity in the gathered data.

cellular phones. However, their research targets were theoretical improvements of, for example, success rate and power consumption with only simulation-based experiments. Implementation work or feasibility study (i.e., *proof-of-concept*) were left untouched.

Though their targets were not primarily on sensor data gathering, UMass DieselNet allowed the study of message routing on bus-based DTNs (e.g., Xiaolan *et al.* [23], MaxProp [3], and RAPID [1]). They used WiFi attached small computers such as DTN-throwboxes [2] for their research. However, we found that most of their work were measurement of contacts and simulation of message routing. Only the paper on RAPID [1] presented the deployment of their DTN routing protocol. But, the scale was not large: i.e., each bus generated four packets (1 k-byte) per hour to every other bus and they achieved 88% success rate. We recognize that we cannot simply compare their work with ours because the assumed application and the network environment are different. However, sensors in our work generated data every 30 s, and our network achieved 99.8% success rate.

Other field-experiments related to sensors were ZebraNet [9], Alan Mainwaring *et al.* [21] and LUSTER [18]. ZebraNet presented a network and system design for wildlife tracking. They developed some prototype systems and planned to deploy 30 nodes to Mpala Research Centre, Kenya. Alan *et al.* carried out field-experiments on habitat and environmental monitoring with Mica Motes. They have shown 28% success rate in multihop cases without DTN. LUSTER deployed static wireless sensor nodes to Hog Island. They applied DTN for improving reliability of communication.

In the context of sensor data gathering, some researchers demonstrate the use of portable sensors: e.g., people-centric sensing [5], BikeNet [8], and vehicular sensing platforms [11]. They say that they can create a map (e.g., CO2 map) with smaller number of portable sensors with the trace of the node movement. The difference is that we have assumed physically fixed sensors in this paper.

An earlier version of this paper [15] has provided a smaller-scale experiment with 11 nodes, which can be called as a preliminary experiment. In this paper, we carried out a larger-scale experiment with 39 nodes. Both experiments have shown great delivery success rate and sufficient usefulness.

VI. Conclusion

In this paper, we have presented our practical study on DTN-based sensor data gathering for agricultural-field sensors. We recognize that cellular and mobile phones could be also used for data collection from remote sensors. However, we demonstrated that DTN-based approach can develop data gathering network by making use of the facilities that farmers or agricultural researchers have: e.g., tractors.

We have carried out an experiment with 39 DTN nodes and PEAR, a DTN implementation, on the campus of the University of Tokyo assuming an agricultural scenario. We confirmed that data from sensors were transferred to the data server with effectively using the movement of vehicles. In our experiment settings, almost all the data (99.8%) were delivered in 10–75 min delay.

References

[1] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "Replication routing in DTNs: A resource allocation approach," *IEEE/ACM Trans. Networking*, vol. 18, no. 2, pp. 596–609, Apr. 2010.
[2] N. Banerjee, M. D. Corner, and B. N. Levine, "Design and field experimentation of an energy-efficient architecture for DTN throwboxes," *IEEE/ACM Trans. Networking*, vol. 18, no. 2, pp. 554–567, Apr. 2010.
[3] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for vehicle-based disruption-tolerant networks," in *Proc. IEEE INFOCOM*, 2006, pp. 1–11.
[4] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-tolerant networking: An approach to interplanetary internet," *IEEE Commun. Mag.*, vol. 41, no. 6, pp. 128–136, Jun. 2003.
[5] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, R. A. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, and G.-S. Ahn, "The rise of people-centric sensing," *IEEE Internet Computing: Mesh Networking*, pp. 30–39, Jul. 2008.
[6] T. Clausen and P. Jacquet, "RFC3626: Optimized link state routing protocol (OLSR)," Oct. 2003.
[7] D. Demmer and K. Fall, "DTLSR: Delay tolerant routing for developing regions," in *Proc. ACM NSDR*, 2007.

[8] S. Eisenman, E. Miluzzo, N. Lane, R. Peterson, G. S. Ahn, and A. T. Campbell, "BikeNet: A mobile sensing system for cyclist experience mapping," *ACM Trans. Sensor Networking*, vol. 6, no. 1, pp. 1–39, Dec. 2009.

[9] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet," in *Proc. ACM SIGOPS*, 2002, pp. 96–107.

[10] F. Laurent and G.-C. Felipe, "Using delay tolerant networks for car2car communications," in *Proc. IEEE Ind. Electron. Symp.*, Jun. 2007, pp. 2573–2578.

[11] U. Lee, E. Magistretti, M. Gerla, P. Bellavista, and A. Corradi, "Dissemination and harvesting of urban data using vehicular sensing platforms," *IEEE Trans. Veh. Technol.*, vol. 58, no. 2, pp. 882–901, Feb. 2009.

[12] X. Li, W. Shu, M. Li, H. Huang, and M.-Y. Wu, "DTN routing in vehicular sensor networks," in *Proc. IEEE Globecom*, 2008, pp. 1–5.

[13] S. M. , X. Li, D. Li, M.-Y. Wu, M. Li, and W. Shu, "Routing in large-scale buses ad hoc networks," in *Proc. IEEE WCNC*, 2008, pp. 2711–2716.

[14] H. Ochiai and H. Esaki, "Mobility entropy and message routing in community-structured delay tolerant networks," in *Proc. ACM AINTEC*, 2008, pp. 93–102.

[15] H. Ochiai, H. Ishizuka, Y. Kawakami, and H. Esaki, "A field experience on DTN-based sensor data gathering in agricultural scenarios," in *Proc. IEEE Sensors*, 2010, pp. 955–958.

[16] C. Perkins, E. Belding-Royer, and S. Das, "RFC3561: Ad hoc on-demand distance vector (AODV) routing," Jul. 2003.

[17] B. Ren, J. Ma, and C. Chen, "The hybrid mobile wireless sensor networks for data gathering," in *Proc. ACM IWCMC*, 2006.

[18] L. Selavo, A. Wood, Q. Cao, T. Sookoor, H. Liu, A. Srinivasan, Y. Wu, W. Kang, J. Stankovic, D. Young, and J. Porter, "LUSTER: Wireless sensor network for environmental research," in *Proc. ACM SenSys*, 2007.

[19] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and Wait: An efficient routing scheme for intermittently connected mobile networks," in *Proc. ACM SIGCOMM*, 2005, pp. 252–259.

[20] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The multiple-copy case," *IEEE/ACM Trans. Networking*, vol. 16, no. 1, pp. 77–90, Feb. 2008.

[21] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An analysis of a large scale habitat monitoring application," in *Proc. ACM SenSys*, 2004.

[22] Y. Wang and H. Wu, "Delay/fault-tolerant mobile sensor network (DFT-MSN): A new paradigm for pervasive information gathering," *IEEE Trans. Mobile Computing*, vol. 6, no. 9, pp. 1021–1034, Sep. 2007.

[23] X. Zhang, J. Kurose, B. N. Levine, D. Towsley, and H. Zhang, "Study of a bus-based disruption-tolerant network: Mobility modeling and impact on routing," in *Proc. ACM MobiCom*, 2007.

Hideya Ochiai received the B.E. degree and M.E. degree in information and communication engineering from the University of Tokyo, Tokyo, Japan, in 2006 and 2008, respectively. Currently, he is working towards the Ph.D. degree at the University of Tokyo.

He is also a Researcher with the National Institute of Information and Communications Technology (NICT), Japan. His research interests cover wide-area sensor networking, delay tolerant networking, and building automation systems.

Hiroki Ishizuka received the B.E. and M.E. degrees from Tokyo Denki University, Tokyo, Japan, in 2006 and 2008, respectively. Currently, he is working towards the Ph.D. degree in information and communication engineering at the University of Tokyo.

His research interests cover sensor networking, application of android phones and participatory sensing.

Yuya Kawakami received the B.E. degree in information and communication engineering from the University of Tokyo, Tokyo, Japan, in 2009. Currently, he is working towards the M.S. degree at the University of Tokyo.

His research interests cover visualization of dataflow in delay tolerant networks, content delivery networks, and interdomain routing.

Hiroshi Esaki received the Ph.D. degree from the University of Tokyo, Tokyo, Japan, in 1998.

In 1987, he joined the Research and Development Center, Toshiba Corporation. From 1990 to 1991, he was with the Applied Research Laboratory, Bell-Core Inc., NJ, as a Residential Researcher. From 1994 to 1996, he was with the Center for Telecommunication Research, Columbia University, New York. Since 1998, he has served as a Professor at the University of Tokyo, and as a board member of WIDE Project. Currently, he is the Executive Director of IPv6 promotion council, Vice President of JPNIC, IPv6 Forum Fellow, and Director of WIDE Project.