A Neighbor Collaboration Mechanism for Mobile Crowd Sensing in Opportunistic Networks

Takamasa Higuchi Hirozumi Yamaguchi Teruo Higashino
Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
{t-higuti, h-yamagu, higashino}@ist.osaka-u.ac.jp

Mineo Takai
University of California, Los Angeles
Los Angeles, CA
mineo@cs.ucla.edu

***Abstract*—Data collection from a crowd of mobile devices is an essential building block of emerging mobile sensing systems. In this paper, we propose an efficient information diffusion protocol for sensor data collection via an opportunistic network. The proposed method detects groups of pedestrians based on the history of radio connectivity between the nodes and maintains a local network (*i.e.*, a cluster) among the detected group members. By collaboratively performing neighbor discovery and link management with the cluster members, it enhances energy-efficiency of the neighbor discovery and minimizes the information delivery delay. Simulation results show that the proposed method can improve the message delivery performance by 16%–83% with equivalent contact probing intervals.**

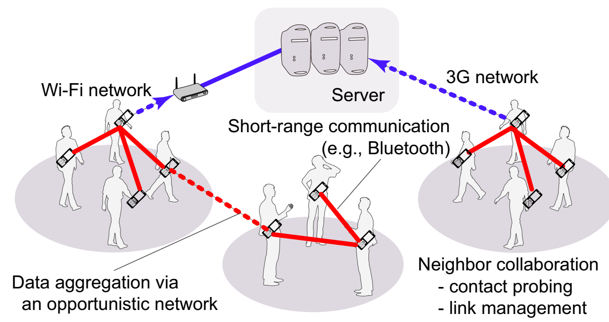

Fig. 1. System Architecture

I. Introduction

Rapid progress in mobile sensing technology has enabled a number of innovative applications and services. By gathering measurement data that are obtained from built-in sensors in mobile devices (*e.g.*, smartphones), we can recognize the current situation and crowd behavior in the environment without densely deploying sensing infrastructure [1], [2]. To provide such mobile sensing applications, we need a scalable platform to collect sensor data from a crowd of mobile devices to servers in the network. Most of existing mobile sensing systems assume that each node uploads its sensor data via cellular networks. However, this approach requires users to pay for 3G communication accompanied by the data uploading, in addition to consuming battery of their mobile devices. In terms of incentivizing people to join the service, this would not be an optimal solution in many participatory sensing scenarios. Now that public IT infrastructure such as free Wi-Fi hotspots is widely available in urban areas (*e.g.*, in train stations and shopping malls), opportunistic networking would provide an alternative solution for data collection in the mobile sensing systems: Each node temporarily stores sensor data in its local storage and uploads them to the server when it comes around the Wi-Fi hotspots. While this may incur some additional delay for the data collection, it would lower the threshold for the users to participate in the sensing service.

A potential problem of this approach is that opportunities for the data uploading are limited since wireless access points such as Wi-Fi hotspots are usually deployed sparsely. To effectively increase the upload opportunities to mitigate the data collection delay, it would be desirable that the sensor data are also shared between the mobile devices. By locally sharing the sensor data among the neighboring nodes via short-range wireless communication and then opportunistically uploading the collected data to a server via 3G or Wi-Fi networks, efficiency of the data collection would be drastically improved (Fig. 1). To exchange the sensor data among a crowd of mobile devices, we need to address the following challenges: The first issue is the trade-off between message delivery delay and energy consumption of the mobile devices. In the situations where people frequently move around, duration of contacts between the nodes tends to be short and often within a few tens of seconds. While each node should frequently perform neighbor discovery to detect such short contacts to reduce data collection delay, it usually incurs heavy energy consumption. The second issue is robustness against node density. In connectionless protocols such as the ad-hoc mode of Wi-Fi, packet reception probability usually declines due to frequent collision of the packets, as the node density increases. In that sense, connection-oriented protocols such as Bluetooth would be more suitable for our case. On the other hand, they often have a constraint on the number of communication links that a node can maintain. When the number of neighboring nodes is much larger than the maximum degree, the node needs to repeatedly establish and disconnect the communication links to the neighbors to exchange sensor data in a timely fashion. Consequently, overhead for link establishment would not be negligible and thus may degrade the bandwidth utilization efficiency. Most of existing routing protocols for opportunistic networks usually assume ideal communication and have not considered such issues in lower layers of the network.

In this paper, we propose an energy-efficient information diffusion protocol for sharing sensor data among the mobile devices over an opportunistic network. In crowded situations like an event place and shopping malls, people naturally form groups and move similarly with some others (*e.g.*, friends, families or strangers who are just moving toward the same destination). The proposed method detects such groups of pedestrians based on the history of radio connectivity between the nodes and maintains a local network (*i.e.*, a cluster) with the group members. By collaboratively performing neighbor discovery and link management with the cluster members, it enhances energy-efficiency of neighbor discovery and copes with the degree constraint of the nodes.

Through simulation experiments using the ONE simulator [3], we compare performance of the proposed method with a non-collaborative approach where all the nodes repeat neighbor discovery with random time intervals and send connection requests to all the detected neighbors. The results show that the proposed method can improve the message delivery performance by 16%–83% with equivalent contact probing intervals.

II. Related Work

A. Routing in Opportunistic Networks

Towards efficient message delivery in an intermittently connected mobile ad-hoc networks, or opportunistic networks, a number of routing protocols have been investigated. The most straightforward approach is epidemic routing [4], in which a node replicates all the messages in its local storage whenever other nodes come within its communication range. While it can achieve minimal message delivery delay without any prior information, it usually incurs an excessive amount of redundant message exchange. A number of works have extended the epidemic routing to cope with this problem by predicting the probability that each neighbor encounters the destination node in the future [5]. Based on the encounter probability, senders make replication decisions for each neighbor to reduce the redundant message transmissions. The routing protocols above basically assume a sparse network where collision of packets rarely happens. On the other hand, in order to apply such routing mechanisms to sensor data diffusion for mobile crowd sensing, it would be essential to (i) address frequent node mobility with minimal energy overhead and (ii) achieve robustness against higher node density. The existing routing protocols do not consider such issues in the lower layers. The proposed method works as a middleware between the routing protocols and Bluetooth MAC/PHY layers to bridge these gaps and achieves efficient sensor data diffusion for mobile crowd sensing via the opportunistic network.

B. Energy-Efficient Neighbor Discovery

Neighbor discovery (or contact probing) has been considered an essential building block of efficient message delivery in an opportunistic network. Wang et al. [6] formulate the contact probing task using renewal theory and proves that constant probing intervals provide an optimal neighbor detection probability, given that the probing is completed instantaneously. eDiscovery [7] achieves energy-efficient neighbor discovery by dynamically adjusting probing intervals based on the number of detected neighbors. Both methods focus on optimizing the probing intervals to avoid *temporal* redundancy, assuming that each node performs contact probing independently. On the other hand, the proposed method collaboratively performs the contact probing among the cluster members to reduce *spatial* redundancy; at each round, only a small number of nodes in the cluster perform neighbor discovery and share the detection results with other members. Since the cluster members are in close proximity with each other, reasonable neighbor detection ratio can be achieved with the reduced probing frequency. Thus the collaborative contact probing mechanism of the proposed method is orthogonal to the existing optimization methods above, so that the efficiency of contact probing could be further improved by adjusting the probing intervals of each cluster based on the temporal optimization strategies in [6], [7].

C. Dynamic Clustering in Mobile Ad-Hoc Networks

A number of clustering protocols have been proposed to reduce routing overhead in large mobile ad-hoc networks. MOBIC [8] finds a node with the smallest relative speed with respect to the neighboring nodes and makes it take on a cluster head to mitigate the frequency of re-clustering due to node mobility. MobHiD [9] predicts future mobility of the nodes based on the history of their contact patterns. Then the node whose neighbor set is assumed to be the most stable in the future takes on the role of a cluster head. While existing clustering algorithms for mobile ad-hoc networks basically assume Wi-Fi or similar protocols for wireless communication, Bluetooth would be a more suitable option for continuous sensor data collection from a number of mobile devices because of its robustness against high node density and lower energy consumption. To strongly support mobile sensing applications, we optimize our protocol for Bluetooth-based communication to cope with its specific constraints, which have not been considered in the existing clustering protocols.

III. Overview

A. Assumptions

In this paper, we assume that people in some public space (*e.g.*, a shopping mall) have mobile devices equipped with ad-hoc communication faculty (*i.e.*, *mobile nodes* or simply *nodes*). To recognize crowd behavior and/or the surrounding situation, sensor data are continuously collected on each node and shared with other nodes in the environment after application-dependent preprocessing. Wireless access points (*e.g.*, Wi-Fi hotspots) are sparsely deployed in the environment, so that the collected sensor data are uploaded to a server in the network when the mobile node comes around the access point. The nodes can also upload its data directly to the server via a cellular network, if the user allows to do so. We also assume that the size of the messages is sufficiently smaller than the wireless network bandwidth. Each node has sufficiently large buffer space and thus buffer overflow never occurs.

B. Overview

The proposed method detects groups of pedestrians based on the recent history of wireless connectivity between the nodes. Then it forms a *cluster* among the detected group members for local collaboration. Each cluster has a special node called *cluster head* (CH), which takes on cluster maintenance and control of intra-cluster communication. Here we consider a CH u_0 and its cluster $\mathcal{M}$. Let $\mathcal{N}_0(t)$ be the neighbor set of u_0 at time t. The proposed method assumes that all the cluster members in $\mathcal{M}$ satisfy the following condition:

$$^{\forall}t' \in (t - T, t];\ u_i \in \mathcal{N}_0(t') \quad (1)$$

Thus a node u_i is required to maintain connectivity with the CH u_0 for more than T seconds to join its cluster. The threshold T should be sufficiently larger than the expected contact duration time for the nodes that are moving independently, which can be estimated via simulations or by analyzing historical trajectory data. Unless otherwise noted, we assume $T = 60$ seconds in this paper. Note that 88% of the contacts between the nodes in different groups are less than this threshold in our simulations in Section VI. Initially, each node forms a cluster with a single member in which the node itself takes on a CH. Afterwards, the CH periodically performs contact probing and updates the member set of its cluster so that all the members satisfy the condition in Eq. (1).

The cluster members continuously maintain a connected intra-cluster network. All the sensor data that are collected by the members are immediately shared with other nodes in the cluster through the intra-cluster communication links. When nodes in different clusters are detected by the neighbor discovery, the node establishes temporary communication links with those neighbors to exchange the messages (*i.e.*, sensor data) between the clusters. While the link establishment is usually accompanied by a few seconds overhead, the message exchange itself can be completed in a short time since the size of the messages is much smaller than the communication bandwidth. Therefore, in the proposed method, cluster members collaboratively manage the inter-cluster links so that at most a single communication link is established between the adjacent clusters. Since the sensor data are immediately shared among the cluster members, the data maintained in the cluster can be transferred to all the nodes in the neighboring cluster through a single communication link. Thus the proposed method mitigates the message delivery delay caused by the degree constraint and the overhead of link establishment by aggregating the inter-cluster message exchange into a small number of communication links.

In addition, the cluster members are spatially in close proximity and thus their neighbor sets would be similar with each other. Based on this feature, the CH randomly selects a small number of cluster members (called *discoverers*) at each round to make them perform neighbor discovery. This effectively mitigates average frequency of the energy-consuming contact probing, while keeping reasonable contact detection capability.

Thus the proposed method mitigates the energy consumption and message delivery delay caused by the degree constraint by (i) forming a cluster with the nodes moving as a group, and (ii) collaboratively performing contact probing, link management and message exchange with the cluster members.

IV. Collaborative Contact Probing

While a node, say u_i, is performing neighbor discovery, it rapidly switches its radio frequency and repeatedly transmits inquiry messages. Accordingly, u_i can hardly respond to the inquiries from neighboring nodes during its neighbor discovery process. Thus detection probability of each neighbor u_j depends on the duration of time when u_j stays in a non-discovery state during the discovery process of u_i itself (*i.e.*, effective scan duration). The neighbor detection probability increases with the effective scan duration T_c and reaches 99% when $T_c = 5.12$ seconds [10]. Although frequent contact probing is required to detect short contacts, it also causes simultaneous neighbor discovery among the neighboring nodes, which incurs missing contacts. In addition, a large amount of battery power is continuously consumed during the discovery process. As mentioned in the previous section, the proposed method copes with the problem by selecting a small number of discoverers and sharing their detection results among the cluster members. This effectively enhances the energy efficiency and responsiveness to the discovery from neighboring clusters.

Let $\mathcal{M}(t)$ be the set of cluster members at time t and $\mathcal{N}_i(t)$ be the neighbor set of a member $u_i \in \mathcal{M}(t)$. For each neighbor $u \in \mathcal{N}_i(t)$, the probability that u is also adjacent to another member u_j would be in proportion to the area of overlapping region of their communication range. Assuming that wireless transmission range is R and the distance between u_i and u_j is given by $r_{ij} = r$, the probability above can be defined as:

$$P(u \in \mathcal{N}_j(t) | u \in \mathcal{N}_i(t), r_{ij} = r) = \frac{1}{\pi R^2}\left\{2R^2 \arccos\left(\frac{r}{2R}\right) - \frac{r}{2}\sqrt{4R^2 - r^2}\right\} \quad (2)$$

Here, we consider the cases where only n cluster members (*i.e., discoverers*) $\mathcal{D}(t) \subseteq \mathcal{M}(t)$ perform neighbor discovery. Assuming that distance between a non-discover u_i and each cluster member follows a probability distribution $f(r)$, the probability that a neighbor of u_i, say u, is discovered by a discoverer u_j in its cluster is given by:

$$\begin{aligned} P(X_{ij}) &= \int_0^\infty f(r_{ij} = r) \cdot P(X_{ij} | r_{ij} = r) dr \\ &= \int_0^\infty f(r_{ij} = r) \cdot P(u \in \mathcal{N}_j(t) | u \in \mathcal{N}_i(t), r_{ij} = r) dr \end{aligned} \quad (3)$$

where X_{ij} represents the event that u_i's neighbor is discovered by u_j. Then we consider the probability that a neighbor of the non-discoverer u_i is discovered by at least one discoverer in the cluster. Let $u_1, u_2, \ldots, u_n$ be the n discoverers in the cluster and $r_{i1}, r_{i2}, \ldots, r_{in}$ denote the distance to each of them. Without loss of generality, we assume that $r_{i1} \le r_{i2} \le \ldots \le r_{in}$. Obviously, the probability $P(X_{i1} \vee X_{i2} \vee \ldots \vee X_{in})$ satisfies the following condition:

$$P(X_{i1} \vee X_{i2} \vee \ldots \vee X_{in}) \ge P(X_{i1}) \quad (4)$$

Note that u_1 is the nearest discoverer from u_i. Let $F(r) = \int_0^r f(r)dr$ be the cumulative distribution of distance from u_i to the discoverers. Then the cumulative distribution function of the distance between u_i and u_1 is given by:

$$\begin{aligned} F_1(r) &= 1 - P(r_{i1} > r \wedge r_{i2} > r \wedge \ldots \wedge r_{in} > r) \\ &= 1 - P(r_{i1} > r)P(r_{i2} > r) \cdots P(r_{in} > r) \\ &= 1 - \{1 - F(r)\}^n \end{aligned} \quad (5)$$

By differentiating $F_1(r)$, we obtain the probability density function of r_{i1}:

$$f_1(r) = \frac{dF_1(r)}{dr} = n\{1 - F(r)\}^{n-1} f(r) \quad (6)$$

Based on the Eq. (3), Eq. (4) and Eq. (6), we obtain:

$$\begin{aligned} &P(X_{i1} \vee X_{i2} \vee \ldots \vee X_{in}) \\ &\ge \frac{n}{\pi R^2}\int_0^\infty \{1 - F(r)\}^{n-1} f(r) \cdot \\ &\qquad \left\{2R^2 \arccos\left(\frac{r}{2R}\right) - \frac{r}{2}\sqrt{4R^2 - r^2}\right\} dr. \end{aligned} \quad (7)$$

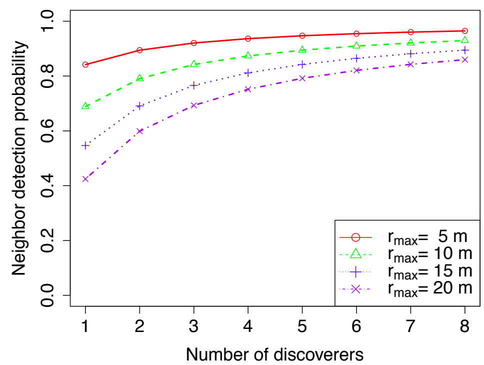

Fig. 2. Neighbor detection probability with collaborative contact probing

To examine the contact detection ratio by the discoverers, we derive the lower bound of $P(X_{i1} \vee X_{i2} \vee \ldots \vee X_{in})$ based on Eq. (7). For the analysis, we assume that the distance between the cluster members follows a uniform distribution between $(0, r_{max})$, where r_{max} is varied between 5m and 20m. The wireless transmission range R is set to 10m based on the typical communication range of Bluetooth. Fig. 2 shows the lower bound of the probability that a neighbor of a non-discoverer is discovered by at least one discoverer in its cluster, as the number of discoverers is varied from 1 to 8. When the maximum distance between the cluster members is 5m, neighbors of non-discoverers can be discovered with probability 84% even if only one member in the cluster takes on the role of a discoverer. Thus, by limiting the number of discoverers in the cluster, we can effectively reduce the average contact probing frequency while keeping reasonable contact detection ratio.

V. Protocol Design

A. Protocol Overview

Initially, each node forms a cluster of size 1. Then each cluster independently repeats the following three phases for cluster maintenance and message exchange.

1) *Cluster maintenance*: The CH performs neighbor discovery to obtain the current neighbor set (Fig. 3 (a)). Meanwhile, other nodes stay in the non-discovery state to respond to the inquiry from the CH. Then the CH updates the members of its cluster based on the collected connectivity information.
2) *Contact probing*: CH randomly selects $(n-1)$ discoverers from the cluster members (except for the CH itself). Then the selected members perform neighbor discovery to detect contacts with neighboring clusters (Fig. 3 (b)).
3) *Message exchange*: The CH and the discoverers establish communication links with the nodes in the neighboring clusters to exchange sensor data that are maintained in each cluster (Fig. 3 (c)). To accept connection requests from the neighboring clusters, the inter-cluster links are immediately disconnected after the message exchange. Finally, the received sensor data are shared among the cluster members through the intra-cluster communication links.

The total duration of each round (consiting of three phases) is randomly determined by CH. Note that the phases 2) and 3) are omitted if a cluster has only one node. In this case, the CH stays in the non-discovery state until the next round starts.

B. Cluster Maintenance

In the cluster maintenance phase, CH, say u_0, turns into the discovery state and broadcasts inquiry messages to the neighboring nodes. When a node u_i receives the inquiry, u_i sends back an inquiry response message that contains IDs (*e.g.*, Bluetooth MAC addresses) of its cluster members. Thus u_0 obtains its current neighbor set $\mathcal{N}_0(t)$ and the cluster members of each detected neighbors: $\{\mathcal{M}_i \mid u_i \in \mathcal{N}_0(t)\}$. Note that $\mathcal{M}_i = \mathcal{M}_j$ if the two neighbors $u_i, u_j \in \mathcal{N}_0(t)$ belong to the same cluster. In the Bluetooth specification, the inquiry response message has an entry for a device name, which is a user-configurable string with the maximum size of 2,048 bytes. By putting the IDs of the cluster members in this entry, the CH can collect the information of neighboring clusters without additional link establishment.

In this phase, only CH performs neighbor discovery and other members stay in the non-discovery state. Thus cluster members can respond to the inquiry from the CH with high probability. On the other hand, the nodes in different clusters may perform neighbor discovery at the same time as the CH since each cluster determines its timing of neighbor discovery independently. Thus some neighbors may be missed depending on the effective scan duration, as mentioned in Section IV. To mitigate the impact of the missing connectivity on the cluster maintenance, the proposed method complements the current neighbor set using the history of connectivity information during the recent W rounds:

$$\mathcal{N}'_0(t) = \mathcal{N}_0(t) \cup \mathcal{N}^c_0(t) \tag{8}$$

where $\mathcal{N}^c_0(t)$ is the set of neighbors that are detected at more than a half of the recent W rounds. While larger W improves robustness of the clustering, it also decreases flexibility against group change. In this paper, we assume $W = 3$ which serves the best performance in our simulation settings in Section VI.

After the neighbor discovery, CH updates its cluster members based on the history of connectivity information. Specifically, it adds and removes the neighboring nodes so that all the cluster members u_i satisfy the following condition:

$$^{\forall}t' \in (t-T, t];\ \ u_i \in \mathcal{N}'_0(t') \tag{9}$$

where T corresponds to that in Eq. (1). Based on the definition above, the CH u_0 updates the cluster members as follows:

1) If a member u_i is not contained in the current neighbor set $\mathcal{N}'_0(t)$, it is detached from the cluster. After that, u_i becomes a CH and forms a new single-node cluster.
2) If all the members in an adjacent cluster $\mathcal{M}_i$ satisfy the condition in Eq. (9), u_0 incorporates all of those nodes into its own cluster to merge $\mathcal{M}_0$ and $\mathcal{M}_i$.

Thus u_0 obtains the current member set $\mathcal{M}'_0$. Finally, the CH u_0 notifies all the cluster members of the updated member set to initiate topology maintenance of the intra-cluster network.

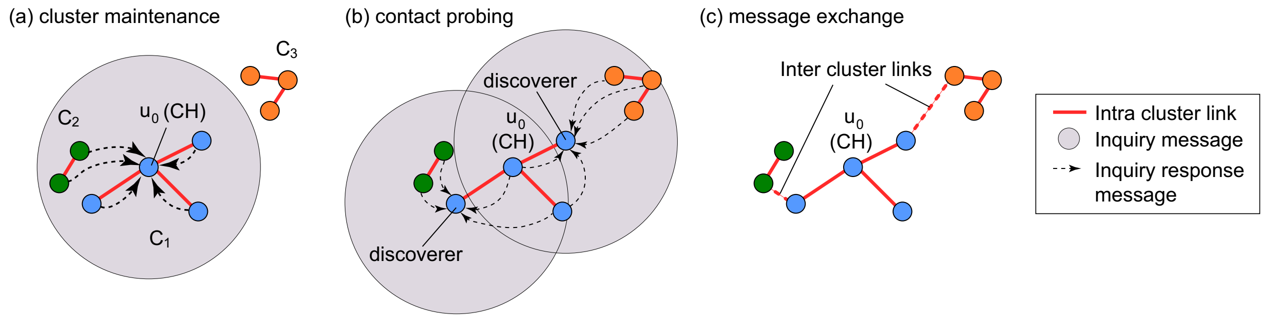

Fig. 3. Protocol Design

Note that the timing of contact probing is independently controlled by each cluster. If two clusters satisfy the criterion in Eq. (9) with each other, the CH that finds it earlier becomes a CH of the merged cluster.

C. Inter-Cluster Communication

In the communication phase, a discoverer $u_i \in \mathcal{D}(t)$ calculates the set of its adjacent clusters based on the current neighbor set $\mathcal{N}_i(t)$ and the cluster member information of each neighbor, in order to form a connection queue $\mathcal{Q}_i = \{\mathcal{M}_j | ^\exists u_j \in \mathcal{N}_i(t) \cap \mathcal{M}_j\}$. Then it randomly selects a member from $\mathcal{N}_i(t) \cap \mathcal{M}_j$ for each adjacent cluster $\mathcal{M}_j \in \mathcal{Q}_i$ to send it a connection request. Meanwhile, the following exclusion control is performed among the discoverers: Before sending a connection request to a neighbor u_k, the discoverer u_i sends a *LOCK*(u_k) message to its own cluster members. This suppresses further connection requests from other discovers to the members in $\mathcal{M}_k$ for a certain period of time. If a communication link with u_k is successfully established, u_i notifies other discoverers of the fact by sending a *CONNECTED*(u_k) message. Then other discoverers remove u_k's cluster from its connection queue. If the connection establishment with u_k is failed, u_i attempts to establish a link with another member in the cluster. Thus a single inter-cluster link is established between the adjacent clusters.

D. Maintenance of Intra-cluster Network

Since Bluetooth has a severe degree constraint, it is desirable that the intra-cluster network can be formed with the minimal number of communication links. To meet the requirement, the proposed method maintains a tree topology in which the CH is a root node. Initially, each cluster forms a tree topology that contains only a root node. When CH u_0 decides to merge an adjacent cluster (let u'_0 be its cluster head), it sends a connection request to u'_0. Then u_0 continues to take on a role of CH in the new cluster. Conversely, if the node is not contained in the cluster member set that is received from CH, it disconnects all the intra-cluster links and becomes a CH of a new single-node cluster. When the communication link with the parent node in the tree topology is disconnected, the member repairs the intra-cluster network by sending a connection request to another member that is not contained in the subtree rooted at the node itself.

When the number of intra-cluster links of a node exceeds a certain threshold, it reconfigures the network topology to increase its available links. If a node u has more than five neighbors, at least two of them are also neighbors with each other [11]. Let v and w be such cluster members. In this case, u instructs v and w to establish a link between them, and then disconnects the link to v (or w). Thus each node can limit the number of intra-cluster links within five to accept connection requests from the nodes in other clusters.

VI. Simulation

A. Simulation Settings

To evaluate performance of the proposed method, we have conducted simulation experiments using the ONE simulator [3]. We assume a field of 50m×50m where 100 mobile nodes randomly move around, forming groups of 10 nodes: We generate 10 reference points each of which represents average behavior of a group. The reference points follow a random waypoint mobility model and move in the field independently. On each occasion of movement, the reference points randomly select a destination waypoint from the field and move toward the destination at a constant speed. The moving speed is also determined randomly from the range between 0.5 m/s and 1.5 m/s. After arriving at the destination waypoint, the reference point stops for up to 120 seconds and then moves toward the next destination. When the reference point determines a new destination, the corresponding group members also pick their own destinations that are within 3m from the reference point. Thus we simulate group-based mobility of the mobile nodes.

Every 10 seconds, each node generates a message of 2Kbytes (*i.e.*, sensor data) and diffuse it towards all the nodes in the environment by epidemic routing. We assume that the deadline of message delivery is 60 seconds and define the message delivery ratio by the percentage of messages that reach the nodes within the deadline. Assuming Bluetooth v2.1+EDR, the range and speed of wireless communication are set to 10m and 2Mbps, respectively. Each cluster performs contact probing at random intervals between $T_{inq} \pm 5.12$ seconds, and the duration time of the neighbor discovery is 5.12 seconds. T_{inq} is a parameter to adjust the average interval of contact probing by each cluster. By default, we set T_{inq} to 15.36 seconds. In the simulation experiments, we determine the neighbor detection probability based on the theoretical model in [10] and limit the maximum number of communication links of each node to up to 7. In addition, we assume an overhead of 2 seconds for the link establishment.

Based on the assumptions above, we ran simulations of 3,600 seconds and compared the performance of the proposed method with a non-collaborative protocol (referred to as *Naive*) where all the nodes repeat device discovery with random

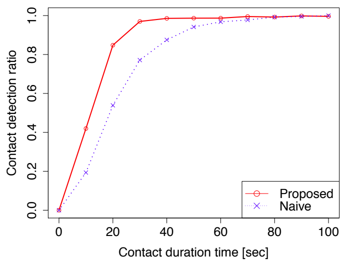

Fig. 4. Contact detection ratio

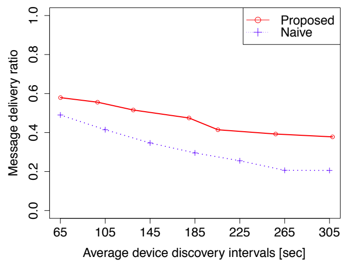

Fig. 5. Message delivery ratio

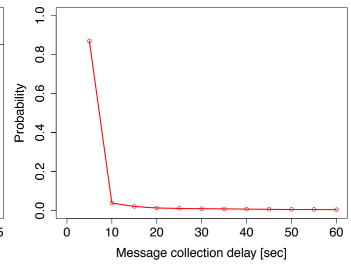

Fig. 6. Message delivery delay

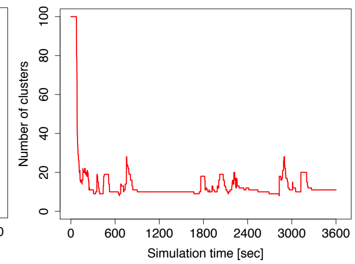

Fig. 7. Number of clusters

time intervals between $T_{inq} \pm 5$ seconds and send connection requests to all the detected neighbors.

B. Simulation Results

1) Contact Detection Ratio: Fig. 4 shows the relationship between contact duration time and contact detection ratio. Since a node can hardly detect the neighbors that are performing neighbor discovery at the same time, some contacts are missed even if all the nodes frequently repeat neighbor discovery with *Naive*. Regarding the proposed method, spatial coverage of contact probing would be slightly lower than *Naive* since it limits the number of discoverers to mitigate energy consumption. On the other hand, it also reduces the simultaneous neighbor discovery among the neighboring nodes and thus the discovers can detect its neighbors with higher probability. As a result, the proposed method could detect short contacts more effectively than *Naive*.

2) Message Diffusion among Mobile Nodes: We also evaluated the message diffusion capability, varying the average contact probing intervals T_{inq}. Fig. 5 shows relationship between the average device discovery interval per node and the message delivery ratio. The proposed method achieves the delivery ratio of 57% with the average device discovery interval of 65 seconds. While longer probing intervals cause missing contacts and thus gradually degrade the message diffusion capability, it still keeps the delivery ratio of 38% with the average probing interval of 308 seconds. Overall, the proposed method could improve the message delivery ratio by 16%–83% by its collaborative contact probing and cluster-based link management mechanisms. Note that the message delivery ratio becomes 73% if both neighbor discovery and message exchange can be completed instantaneously (*i.e.*, under ideal communication). This would be a performance upper bound with this simulation scenario.

3) Data Collection to a Server: Finally, we evaluate the message collection delay by a server, assuming a member in each cluster uploads the received messages via 3G networks. Fig. 6 and Fig. 7 show the probability distribution of the delay time and the number of clusters through the experiment, respectively. Initially, the number of clusters is 100 since each node forms a single-node cluster. After that, the clusters gradually grow by merging the neighboring clusters, and eventually the number of clusters converges between 10 and 20. Since the sensor data collected by each node are immediately shared with other cluster members, more than 87% of the messages reach the server within 5 seconds. Considering that the average number of clusters throughout the simulation is 14.3, the number of communication links to the server is effectively reduced by 86% compared with the centralized cellular-based architecture, while maintaining a reasonable data collection performance. Thus the local message exchange between the mobile nodes effectively enhances the scalability of the mobile sensing system.

VII. Conclusion

In this paper, we have proposed an energy-efficient information diffusion protocol for mobile crowd sensing in an opportunistic network. The proposed method detects groups of pedestrians to form clusters, in which the nodes perform link management and contact probing in a collaborative manner. Simulation results have shown that the proposed method could improve the message delivery ratio by 16%–83% with equivalent contact probing intervals.

Acknowledgment

This work was supported in part by the KDDI foundation and the CPS-IIP Project (FY2012 - FY2016) in the research promotion program for national level challenges by the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan.

References

[1] J. Weppner and P. Lukowicz, "Bluetooth based collaborative crowd density estimation with mobile phones," in *Proc. PerCom '13*, 2013, pp. 193–200.

[2] L.-Y. Wei, Y. Zheng, and W.-C. Peng, "Constructing popular routes from uncertain trajectories," in *Proc. KDD '12*, 2012, pp. 195–203.

[3] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation," in *Proc. SIMUTools '09*, 2009.

[4] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," Tech. Rep., 2000.

[5] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, "MaxProp: Routing for vehicle-based disruption-tolerant networks," in *Proc. INFOCOM '06*, 2006, pp. 1–11.

[6] W. Wang, V. Srinivasan, and M. Motani, "Adaptive contact probing mechanisms for delay tolerant applications," in *Proc. MobiCom '07*, 2007, pp. 230–241.

[7] B. Han and A. Srinivasan, "eDiscovery: Energy efficient device discovery for mobile opportunistic communications," in *Proc. ICNP '12*, 2012, pp. 1–10.

[8] P. Basu, N. Khan, and T. Little, "A mobility based metric for clustering in mobile ad hoc networks," in *Proc. ICDCS Workshop*, 2001, pp. 413–418.

[9] C. Konstantopoulos, D. Gavalas, and G. Pantziou, "Clustering in mobile ad hoc networks through neighborhood stability-based mobility prediction," *Comput. Netw.*, vol. 52, no. 9, pp. 1797–1824, 2008.

[10] B. Peterson, R. Baldwin, and J. Kharoufeh, "Bluetooth inquiry time characterization and selection," *IEEE Transactions on Mobile Computing*, vol. 5, no. 9, pp. 1173–1187, 2006.

[11] G. Zaruba, S. Basagni, and I. Chlamtac, "Bluetrees – scatternet formation to enable bluetooth-based ad hoc networks," in *Proc. ICC '01*, vol. 1, 2001, pp. 273–277.