A Method for Updating Location-based Information on Ad-hoc Network

Yoji ISHII, Koichi ASAKURA, and Toyohide WATANABE
Department of Systems and Social Informatics,
Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-8601, JAPAN
E-mail: {yoji,asakura,watanabe}@watanabe.ss.is.nagoya-u.ac.jp

Abstract *We propose a method which can consistently update location-based information kept by mobile terminals which are located in a position where location-based information is provided and which communicate with each other on ad-hoc network. In our approach, mobile terminals elect a mobile terminal as a coordinator providing the consistency control mechanism. We propose an election algorithm for a coordinator among indefinite mobile terminals. In addition, we propose a negotiation protocol in which two coordinators specify one of them as a new coordinator when more than one coordinator is elected caused by unstable wireless communication link. The results of our simulation experiments show that our proposed method can update information around 2 seconds and elect a coordinator around 5 seconds.*

Keywords ad-hoc network,election algorithm, location-based information,network protocol

1 Introduction

Location-based information services (LBS) have been getting remarkable attention[1]. In the LBS, information is associated with a position and provided for the users in the associated position[2, 3]. We have proposed a method which can realize the LBS without any centralized servers[4]. Namely, data for location-based information is kept by mobile terminals located in the associated position, and these mobile terminals communicate with each other using ad-hoc network in order to manage the location-based information. By adopting above approach, information which must be sent immediately and whose life time is very short can be dealt with adequately since all operations on data for location-based information can be performed in the local position.

In our approach, data for location-based information is kept by plural distributed mobile terminals. Therefore, if data is updated, an update control mechanism is required in order to keep consistency of location-based information[5]. For this requirement, we adopt a method in which a mobile terminal controls update sequences of mobile terminals exclusively as a coordinator. In order to realize the above mechanism, a distributed election algorithm has to be provided. However, it is difficult to elect a mobile terminal because candidate mobile terminals located in the associated position are indefinite and a wireless communication link is unstable in the ad-hoc network.

In this paper, we propose an election algorithm in which indefinite mobile terminals can elect a mobile terminal as a coordinator. In the proposed algorithm, each candidate mobile terminal calculates a candidate value by which suitability as the coordinator is expressed using only local information regarding to network resources. By using local information, all indefinite mobile terminals in the associated position can be identified, and an appropriate mobile terminal can be elected as the coordinator. Moreover, we propose a negotiation protocol in which two coordinators can specify one of them as a new coordinator in case that more than one coordinator is elected caused by a unstable wireless communication link.

The rest of this paper is organized as follows. In Section 2, we present a consideration when location-based information is updated in our assumed environment. In Section 3, we describe a method which realizes LBS with update control mechanism. In Section 4, we present simulation experiments. In Section 5, we conclude this paper and describe our future work.

2 Consideration of Updating Location-based Information

In our approach, one mobile terminal has to be elected as a coordinator for updating location-based information. Furthermore, this election has to be achieved by mobile terminals themselves. Then, the coordinator controls the update sequences of mobile terminals exclusively to keep consistency of information. Therefore, a distributed election algorithm must be provided. The bully algorithm is a typical distributed election algorithm for electing one coordinator system among plural systems[6]. The bully algorithm assumes that each system has its own priority number and can send a message to all of other systems unfailingly. In the algorithm, priority numbers are exchanged, and the system with the highest priority number is elected as the coordinator system. However, this algorithm cannot be adapted for our assumed environment. Since the mobile terminals move into and out the associated position dynamically, we cannot assign priority numbers to them. In addition, the wireless communication link is unstable: the network is frequently divided caused by movement of mobile terminals, and messages may not be reached to mobile terminals in its transmission range caused by collisition of wireless network. As a result, more than one mobile terminal may be elected as a coordinator.

From the above consideration, we have to take the following two topics into account: adequate priority number has to be assigned to mobile terminals and a negotiation protocol to specify one coordinator has to be provided in case that plural coordinators are elected accidentally. For the first problem, we require a method which can order indefinite mobile terminals located in the associated position without the centralized server. In our approach, we calculate a candidate value by which suitability as the coordinator is expressed using a position of a mobile terminal and an elapsed time from moving to the associated position. Since the position and the time are peculiar to each mobile terminal and acquired by themselves, we can use these information to order mobile terminals for the election. The election algorithm is invoked whenever the coordinator moves out from the associated position. If the election algorithm is invoked at many times, the network resources are consumed. Therefore, we require to decrease the frequency of algorithm invocation. For this problem, we propose a calculation formula which assigns higher candidate values to mobile terminals which will stay in the associated position for a longer time.

For the second problem, we require a method which can specify a coordinator among plural coordinators. It is important to take account of that mobile terminals in each divided network may update information independently and different versions of information may be created. Therefore, the different information must be transmitted to the specified coordinator surely. Thus, we propose a negotiation protocol in which one mobile terminal is specified from two coordinators and their information is exchanged surely.

3 LBS Using Ad-hoc Network with Update Control

In our method to realize the LBS, mobile terminals located in the associated position keep location-based information without any centralized servers. Therefore, the mechanism to transmit the information immediately to a mobile terminal which moves into the associated position is required.

3.1 Community

In order to realize the LBS by only mobile terminals, we introduce a concept of a community. We define a community area (CA) as the associated position where the location-based information is provided. Then, mobile terminals in the CA form a community and manage data for location-based information using our proposed ad-hoc network protocol called a *community protocol*[4].

In the community protocol, each member of a community sends a beacon packet periodically in order to notify an existence of a community to mobile terminals in its transmission range. In the beacon packet, the CA information is included. When a mobile terminal receives a beacon packet, it evaluates whether it is in the CA or not. If new CA information is included in the beacon packet, a mobile terminal sends a request packet to the sender in order to get the community information. This sequence to acquire the community information from a neighboring member is called an *acquisition sequence.*

3.2 Election Algorithm for Coordinator in Community

All mobile terminals which move into the CA can be members of the community. In this case, members of the community are indefinite. Namely, many types of mobile terminals may be community members, and the number of community members is varying. Thus, it is impossible to assign priority numbers to all of members to order them. In order to solve this problem, we propose a method which uses a candidate value (CV) by which suitability for the coordinator is expressed to order members in the community. Here, the suitable coordinator is defined as a member which will work as the coordinator for a longer time. The suitable coordinator has the following properties:

- it exists in the community for a longer time, and
- it locates near center of CA.

The first property comes from an assumption that the member in the community for a long time will exist in the community hereafter. The second property comes from the fact that the member which locates near center of CA will take time to move out from the CA and that it may locate center of the network topology and response time from the coordinator may become shorter. Therefore, the formula to calculate CV is defined as follows:

$$CV = \alpha \times \log(d) + (1 - \alpha) \times \log(t)$$

where α is a constant ($0 < \alpha < 1$), d is a distance from a current position of the member to an edge of the CA, and t is an elapsed time from being the member of the community. We use a logarithmic function for d because a member moving at high speed near center of the CA may leave a community in a short time and we require to prevent such a member from getting high CV. In addition, we use a logarithmic function for t in order to prevent a member which is in the community for a long time but locates at an edge of the CA from getting high CV. Using above formula, each member can calculate its own CV using local information.

When the coordinator leaves the community, an election algorithm is activated. In the algorithm, each mobile terminal calculates its own candidate value cv and broadcasts cv as an *election packet.* If it receives an election packet from other members, cv in the packet is examined whether the value is higher than its own cv or not. If the value of the cv in the packet is higher, this terminal never be a coordinator. If the election packet in which higher value of cv exists never be received during a timeout, $T_{candidate}$, this member becomes a coordinator of the community. Figure 1 shows an election algorithm in detail.

In order to start the election algorithm, each member must sense absence of the coordinator. In our approach, the coordinator notifies its existence to all members. Namely, the coordinator periodically sends a packet called a *de-*

```
INPUT: cv_k //received as election packet
OUTPUT: cv //broadcast as election packet

INISIALIZE:
  state=NON-ELECTION;

begin
    if (state=NON-ELECTION) then
        cv_i=getCV();
        if (cv_i > cv_k) then
            state=CANDIDATE;
            return cv_i;
        else
            state=NON-CANDIDATE;
            return cv_k;
        endif
    else if (state=CANDIDATE) then
        if (cv_i < cv_k) then
            state=NON-CANDIDATE;
            return cv_k;
        endif
    else if (state=NON-CANDIDATE) then
        if (cv_i < cv_k) then
            return cv_k;
        endif
    endif
end
```

Figure 1: An election algorithm invoked by a member M_i receiving cv_k

terrence packet to all members in the flooding manner. When a member has not received the deterrence packet during a timeout, $T_{deterrence}$, it considers the coordinator leaves the community and starts the election algorithm. In addition, the deterrence packet is used to notify to be elected in the election algorithm.

3.3 Negotiation Protocol between Two Coordinators

Because of the unstable communication link, more than one coordinator may exist in the same community. When the network division occurs, members cannot receive the deterrence packet from the coordinator. Then, they start the election algorithm and elect one member as a new coordinator among them. In addition, when a member fails to receive the election packet containing higher CV caused by collision of wireless network, the member elects itself. Thus, even in the same community, more than one coordinator may be elected.

In order to deal with the above problem, we propose a method for specifying one coordinator from two coordinators. When a coordinator detects other coordinators, an *exchange sequence* shown in Figure 2 is invoked by sending an *exchange packet*. Namely, when the coordinator C_i receives the deterrence packet from the other coordinator C_j, C_i and C_j exchange their *cv* as the exchange packet in the flooding manner, and they specify which coordinator is better to be a new coordinator. In this exchange sequence, location-based information must be sent to the specified coordinator since each coordinator may have independently updated data. Therefore, in the exchange sequence, location-based information is also exchanged by describing in the exchange packet in addition to *cv*. In addition, this sequence must be achieved precisely since two coordinators exchange location-based information. Therefore, they exchange an end packet each other to close the exchange sequence surely.

In our method, we define that data for location-based information kept by the coordinator is the one which should be managed in the community. Therefore, when the new coordinator is elected in the election algorithm or specified in the exchange sequence, members must acquire the new location-based information. In order to realize it, each member contains *coordinator* as the coordinator ID which the member belongs to in the beacon packet. On receiving a deterrence packet from other coordinator C_j, a member invokes a *new network sequence* and waits in $T_{newnetwork}$ in order to acquire the new information. During $T_{newnetwork}$, if the member receives the beacon packet containing C_j as the coordinator ID, the member invokes the acquisition sequence in order to acquire new data of the coordinator C_j.

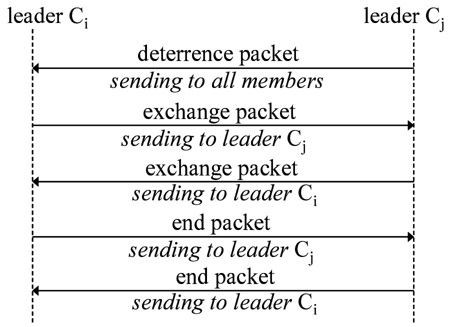

Figure 2: Sequence diagram for specifying one coordinator from two coordinators

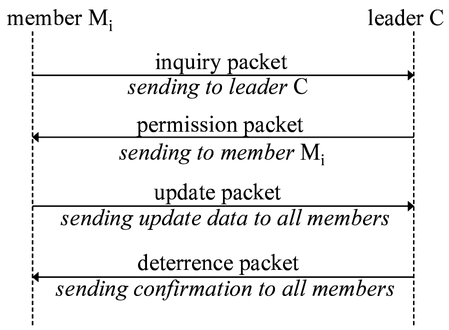

Figure 3: Sequence diagram for updating location-based information

3.4 Updating Location-based Information

In order to update location-based information exclusively among members, a member must acquire a permission for update from the coordinator. First, a member invokes an *update sequence* shown in Figure 3 by sending a packet called *inquiry packet* to the coordinator in the flooding manner in order to acquire the permission. If the coordinator does not give the permission to other members, it sends a packet called *permission packet* to the member in the flooding manner. On receiving the permission packet from the coordinator, the member sends a packet called *update packet* in which update data is described to all of other members in the flooding manner.

In our method, we define that data for location-based information kept by the coordinator is the one which should be managed in the community. Therefore, all members which receive the update packets must wait to update data in their local memories until they confirm that the coordinator receives the update packet. In our method, the coordinator describes *ver* as the version number of its data for location-based information in the deterrence packet, and a member can confirm if the coordinator receives the update packet by comparing *ver* in the deterrence packet with the version number of update data.

In the above update sequence, there may be some members which cannot receive the update packets because of the unstable communication link. In order to deal with such situation, members describes *ver* as the version number of data in the beacon, and a member receiving it invokes the acquisition sequence if *ver* is higher than its version number of data.

4 Simulation Experiments

In order to evaluate our method, we conduct two sets of simulation experiments. We evaluate time to update data for location-based information and time to elect a coordinator. In these simulations, we assume vehicles as mobile terminals since our proposed method is more effective to provide drivers with safety information. All mobile terminals are allocated randomly on 5 × 5 grid roads each of whose length is 120 m. They move at 40 km/h and turn randomly on intersections.

Parameters for simulations are shown in Table 1. We define sizes of all packets including protocol head as 50 bytes. This number is enough to evaluate our protocol since our protocol deals with control data which is much lower than 50 bytes. We define communication latency between two mobile terminals as 1 [ms]. When a mobile terminal receives more than one packet within 1 [ms], it deletes these packets in order to model collision of wireless network.

Parameters for protocol are shown in Table 2. In these experiments, one community is created in the center of the simulation field. We set 1 [sec] for interval to send beacon packets and 2 [sec] for interval to send deterrence packets. We use 0.8 in formula for CV since we

Table 1: Parameters for simulation

communication range	100 m
packet size	50 bytes
communication latency	1 ms
link speed	400 kbps

Table 2: Parameters for protocol

radius of CA	150 m
$INTERVAL_{beacon}$	1 [sec]
$INTERVAL_{deterrence}$	2 [sec]
$T_{candidate}$	1 [sec]
$T_{deterrence}$	4 [sec]
$T_{newnetwork}$	1.5 [sec]
α in CV	0.8

Table 3: Simulation information

Num of terminals	100	200	300	400	500	600	700
Num of members	17	35	54	73	89	107	126
Ave of neighbors	6	10	17	22	27	33	38

consider the location is more reliable for time to being in a community.

For each experiment, we allocate 100 to 700 mobile terminals increasing by 100 in the simulation field. For each total number of mobile terminals, the average number of members and the average number of neighboring mobile terminals are shown in Table 3. We evaluate 100 times for each total number of mobile terminals and calculate average of update time and election time.

4.1 Results of Experiments

In the experiment to evaluate the update time, a member selected randomly updates data for location-based information. We measure time from the selected member sends the inquire packet to all members complete updating data. In order to complete updating data, there are two ways. One is by receiving an update packet from the selected member and a deterrence packet from a coordinator, and the other is by failing to receive the update packet from the selected member and acquiring the update data using the acquisition sequence (Section 3.4). In this experiment, we do not perform the election algorithm. In the experiment to evaluate the election time, we measure time from a member sends the election packet to a coordinator is elected and all members receive data for location-based information which a new coordinator keeps by invoking a new network sequence (Section 3.3).

Figure 4 shows the average update time and the average election time. When the total number of mobile terminals is 100, the update time is 2.8 [sec], and the election time is 7.1 [sec]. Since members exist sparsely, the network is frequently divided. In the update sequence, some members cannot receive update data and acquire it in the acquisition sequence after the network is connected. This causes increase of the update time. In the election algorithm, plural members are elected in each divided network. In this case, the exchange sequence is invoked when two coordinators meet each other. This causes increase of the election time.

When the total number of mobile terminals is 400 to 700, the update time and the election time increase gradually. When the number of members increases, the collision of wireless network occurs frequently. In the update sequence, the number of members which cannot receive the update packets and acquire update data in the acquisition sequence increases. In the election algorithm, the number of members which cannot receive the election packets and elect themselves increase. As a result, the exchange sequences are invoked many times, and this causes increase of the election time. In addition, in a situation that many coordinators exist, they send exchange packets to coordinators which send other coordinators, and these coordinators come to a deadlock. In this case, they must wait until the timeout for the exchange sequence exceeds, and it takes time to specify one coordinator.

When the total number of mobile terminals is from 200 to 400, the update time is around 2 [sec], and the election time is around 5 [sec]. In

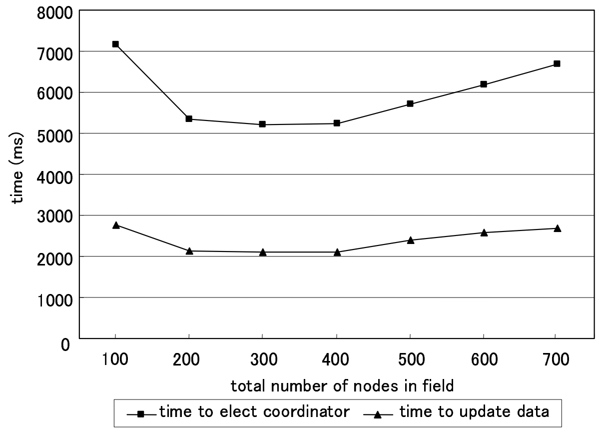

Figure 4: Election time and update time vs. number of mobile terminals

the update sequence, since the inquiry packet and the permission packet are exchanged between the selected member and the coordinator, and the update packet and the deterrence packet are sent to all members, 2 [sec] are reasonable time in our method. In the election algorithm, only one coordinator is elected at the $T_{candidate}$ most times. However, it takes time to acquire the data which the new coordinator keeps using the new network sequence, and the election time increases more than $T_{candidate}$.

As a result of these experiments, even in an environment that an average neighbors is 10 to 22 (Figure 3) like urban area, our proposed method consistently updates location-based information within one figure, and drivers can acquire safety information about surrounding situation which changes dynamically.

5 Conclusion

We described an algorithm which can elect a coordinator among indefinite mobile terminals in order to update location-based information kept by them consistently. In the algorithm, each mobile terminal calculates a candidate value by which suitability for the coordinator is expressed using local information: a position of the mobile terminal and an elapsed time from it moves into the associated position. In addition, we propose a negotiation protocol which can specify a coordinator from two coordinators in order to deal with the situation that plural coordinators exist in the associated position because of the unstable communication link. The simulation results show that our proposed method can update data around 2 [sec] and elect a coordinator around 5 [sec]. We can clarify that our proposed algorithms make it possible to share and update location-based information rapidly and consistently.

In future work, we must propose a method which can specify a coordinator among more than two coordinators. Since our proposed method is invoked between two coordinators, coordinators may come to a deadlock when more than two coordinators exist.

Acknowledgements

The authors would like to thank the 21^{st} Century COE (Center of Excellence) Program for 2002, a project titled Intelligent Media (Speech and Images) Integration for Social Information Infrastructure, proposed by Nagoya University.

References

[1] E. Kaasinen. *User requires for location-aware mobile services.* Personal and Ubiquitous Computing Magazine, vol. 7, no. 1, pp. 70-79, 2003.

[2] H. Tarumi, K. Morishita, M. Nakao, and Y. Kambayashi. *SpaceTag: An Overlaid Virtual System and its Application.* Proc. of Int'l Conf. on Multimedia Computing and Systems (ICMCS'99), Vol. 1, pp. 207-212, 1999.

[3] H. Tarumi, K. Morishita, and Y. Kambayashi. *Public Applications of SpaceTag and their Impacts.* Digital Cities: Technologies, Experiences and Future Perspectives, Lecture Notes in Computer Science, Vol. 1765, pp. 350-363, 2000.

[4] Y. Ishii, K. Asakura, T. Watanabe. *Location-based Information Services Using Ad-hoc Network.* Proc. of 10th Int'l Conf. on Virtual Systems and Multimedia, pp. 1084-1092, 2004.

[5] U. M. Borghoff, J. H. Schlichter. *Computer-Supported Cooperative Work ch. 5.* Springer, 2000.

[6] J. L. Peterson, A. Silberschatz. *Operating System Concepts. 2nd ed. ch. 13.* Addison-Wesley, 1985.