A Geocast scheme based on Successful Transmission Records on Wireless Ad Hoc Networks

Akihiro Nomoto†, Gen Tsuchida‡and Susumu Ishihara‡

†Graduate School of Science and Engineering, Shizuoka University, nomoto@ishilab.net
‡Graduate School of Science and Technology, Shizuoka University, {gen, ishihara}@ishilab.net

ABSTRACT

In this paper we propose an efficient geocasting scheme based on flooding for cases where a source node sends request messages to nodes in the destination area and receives replies from some of them, Geocasting based on Successful Transmission Records (GSTR). Generally geocast protocols based on flooding generate many redundant messages which generate transmission errors. In GSTR, each node records whether it has forwarded a reply message for a query message that the node forwarded to the destination region. Using these records, each node decides the delay before rebroadcast of geocasting a message to avoid forwarding of redundant query messages and forward the message along a suitable path by priority.

Simulation results showed that the proposed scheme reduces collisions and response time to queries, and achieves high success ratio.

Keywords: Ad-hoc networks, Location-based services, Sensor networks, Geocast, Successful transmission record

1 INTRODUCTION

Recently, services taking location information are widely deployed with wide spread of GPS. For example there are car navigation systems with VICS (Vehicle Information and Communication System) and pedestrian navigation systems distributing local information. Management of geographical information and local information has been attracting research interest.

Services sharing location dependent information on ad-hoc networks composed only of mobile nodes is highly expected, because they do not require large and costly infrastructure. For example, services that exchange location-dependent information such as traffic information or bargain sale information between mobile phones or vehicles can be achieved using this advantage. We call such a system SOLA (System for Sharing Objects with Location information on Ad-hoc networks) Fig.1). We assume such location dependent data handled by SOLA are generated by mobile nodes and associated with the position where the data item was generated. We have proposed techniques for distributing replica of location-dependent information to improve the availability of data items generated by each node on ad hoc networks where the connectivity between nodes is not guaranteed [1][2].

On SOLA, every node which requests location-dependent information does not know which node has a target data item. Instead of using fixed servers that know hosts that have data items, each node sends a query message for a target data item to nodes near the geographical area where the data items is related to by geocast, assuming that nodes near the related location have the replica or original of the target data item. Nodes that have the replica or original of the data item reply to the requesting node. This paper discusses the optimization of geocast on such a situation where nodes send queries for location dependent information by geocasting and nodes which have the requested data send back to the requesting node.

So far various ad-hoc routing protocols have been proposed, and most of them are designed to route packets to nodes which are identified by their addresses. However on SOLA, each node uses geocast for sending queries to obtain location dependent data items to multiple nodes that exist near the related to the target area.

Many geocasting schemes for ad hoc networks have been proposed. They can be classified into unicast-based schemes and flooding-based schemes Fig.2). Unicast-based schemes usually use hello-packets to exchange node's location information, and each intermediate node selects a next node and forwards the packet to the node. Repeating the same routine and flooding packets in the target geocast region, packets are delivered to all nodes in the geocast region. On the other hand, in flooding-based schemes, a source node sends packets to all nodes in a flooding region which covers the source node and the geocast region using flooding limited with location information. However flooding-based schemes generate a lot of tedious packets. Therefore collisions occur frequently because of a large amount of packet sent from multiple neighboring nodes, and it may decrease the reachability of the packets the destination. Accordingly it is important to inhibit traffic to avoid collisions.

In SOLA, query packets should be delivered to nodes that have the replica or original of the target data item. Such nodes may not exist in the geocast region which covers the related location where the target data item was generated. If a node has obtained a data item generated at a location, the node may leave there after a while. Therefore, there may be no nodes having the replicas (or original) of data item related to the location at the location even if the replicas of the data item were distributed around the location by a scheme presented in [1]. Thus to use flooding-based geocasting protocols are useful in such a situation because the reachability of request message will be improved by deliveries of query message to nodes outside geocast region using flooding-based geocast.

In this paper we propose an efficient flooding-based geo-

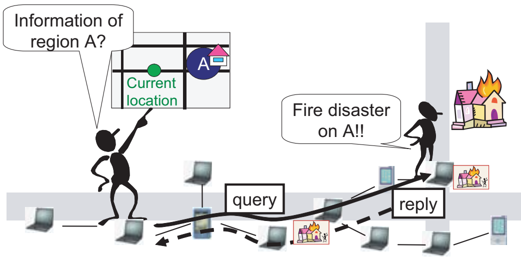

Figure 1: System for Sharing Objects with Location information on Ad-hoc networks (SOLA)

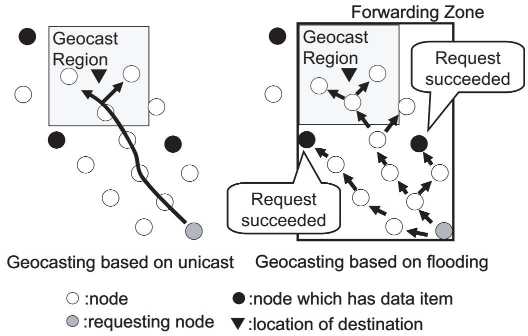

Figure 2: Forwarding queries by geocast on SOLA

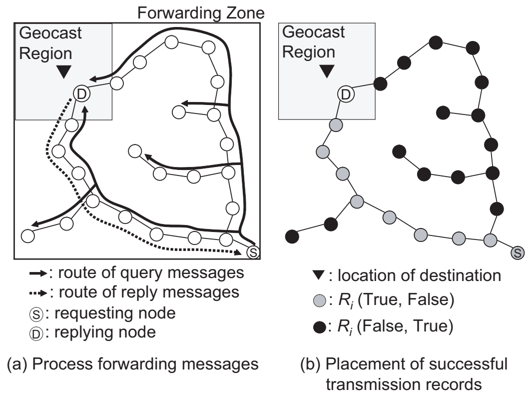

Figure 3: Obtaining R_i in GSTR

cast scheme: Geocast based on Successful Transmission Records (GSTR). The remainder of this paper is organized as follows. In Section 2, we briefly review related work in geocast and optimization of broadcast on ad hoc networks. Then, the detail of GSTR is presented in Section 3. Section 4 gives the detail of the simulation. Then the results and discussion are presented in Section 5. Finally we conclude this paper in Section 6.

2 RELATED WORK

2.1 GEOCAST BASED FLOODING

Location Based Multicast (LBM) is a well known flooding-based geocast scheme[3]. It defines a forwarding zone that comprises a subset of all network nodes. The forwarding zone includes at least the target area and a path between the sender and the target area. An intermediate node forwards a packet only if it belongs to the forwarding zone. A node broadcasts a received packet to all neighbors provided that this packet was not already received before and that the node belongs to the forwarding zone. Finally, a node accepts a packet and delivers it to its application if the own location is within the specified target area. However, because all nodes in the forwarding zone transfer geocast packets, there may be a large amount of redundant packets in case of the distribution is dense.

2.2 OPTIMIZATION OF BROADCAST

Ni identified the problem of broadcast storm and proposed the following five schemes that alleviate this problem. i) Nodes transfer packets by probabilistic rebroadcast, ii) Forwarding nodes insert random delay before rebroadcast, and if the number of packets received before transfer is over a threshold, nodes inhibit to rebroadcast. iii) Nodes suppose already covered area by the distance to the nearest node which the same message is heard before a rebroadcast message is actually sent, and decide to whether to rebroadcast or not. vi) Nodes check already covered area using the locations of nodes which rebroadcasted geocast messages and avoid duplicate rebroadcast, v) Nodes check neighbor nodes using hello packets and nodes which belong to multiple cluster broadcast packets[4]. Fasolo proposed a scheme to control delay of rebroadcast the most distant from transferred nodes[5]. OLSR, a proactive routing protocol for MANET, is designed to reduce flooding overhead using MPR flooding[6].

3 A GEOCAST SCHEME BASED ON SUCCESSFUL TRANSMISSION RECORDS

On SOLA, all nodes relay queries by LBM. All nodes in forwarding zone forward query messages. Then, nodes which have a requested data item send back reply messages along the reverse of the route where the query messages were forwarded.

In GSTR, each node records whether the node has forwarded a reply message after it forwarded the query message. When a node receives a query message, it inserts a random delay, *pre-transmission delay*, calculated with the records —Successful Transmission Records— before rebroadcasting the message, so that a route which was used for successful query is used by priority. Thus, GSTR can avoid collisions of packets by concurrent rebroadcast by multiple nodes that receive a query message and reduce response the query. In addition, a node A which is waiting for the finish of pre-transmission delay cancels to forward a query message if it receives the same message forwarded by nodes whose two-hop neighbor is a node that sent the same message to node A

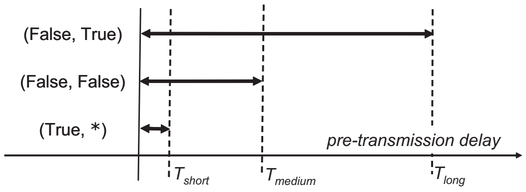

Figure 4: pre-transmission delay on GSTR

directly. As a result, redundant messages are reduced.

3.1 GETTING SUCCESSFUL TRANSMISSION RECORDS

The successful transmission records are obtained by nodes which transfer queries when they forward reply messages, so they can obtain successful transmission records without additional packets. We assume a whole field is divided to n sub-areas. Each node maintains a successful transmission record R_i for each sub-area $i\,(=1, 2, \cdots, n)$. R_i is presented as (r_s, r_f). r_s represents whether a node has forwarded a reply message after it forwarded a query message. r_f represents whether a node has never forwarded a reply message after forwarded a request message. r_s and r_f are initialized to *False*. When a node forwards a query message to a sub-area i, it sets an expiration time T_e of the query for R_i. When a node forwards a reply message to the query before the expiration, it sets r_s of R_i to *True*. If the expiration time elapses before the node forwards the reply message to the query, r_f of $_i$ is set to *True*. T_e is calculated as a function of the passing speed v of a node.

$$T_e = \min(k_r/v, T_0), \tag{1}$$

where k_r is a positive constant value, and T_0 is the maximum value of T_e.

3.2 CONTROLLING TIMING OF REBROADCATST WITH SUCCESSFUL TRANSMISSION RECORDS

In GSTR, when a node has to forward a query message, it inserts a random pre-transmission delay, before the rebroadcast of the query message. The range of the pre-transmission delay $[0, T_{\max}]$ is calculated with successful transmission records R_i for geocast region i. When R_i is (*True*, *), the node uses a short maximum delay. That's T_{max} = T_{short}. When R_i is (*False*, *False*), it uses a medium maximum delay T_{medium}, and when R_i is (*False*, *True*), it uses a long maximum delay T_{long}. Of course, $0 \leq T_{short} \leq T_{medium} \leq T_{long}$.

3.3 CONCENTRATION OF NODES MAINTAINING RECORD AROUND DESTINATION

Because nodes around a destination area of a query message are likely to reach to a node which has target data item, they often obtain a successful transmission record (*True*, *) for the destination. Thus many nodes forward packets using short pre-transmission delay. As a result, many packets are lost because of many collisions due to multiple transmissions of the same message from the neighboring nodes.

To solve this problem, we propose a scheme to decide the pre-transmission delay by considering the distance between a forwarding node and the center of the destination area of a query message. If the distance is smaller than the threshold D_{near}, the node uses medium maximum delay T_{medium} even though it has $R_i = (\mathit{True}, *)$. Fig.6 shows the range of the pre-transmission delay used in this enhanced scheme.

```
A) When a new query message "query" arrives:
  if (I have the requested data item) {
    sendReply();
  } else if (I'm in ForwardingZone) {
    unrepliedQueries.add(query, moving_speed);
    scheduleQueryExipiration(query);
    // After the expiration, go to process C.

    transmissionRecord record
      = transmissionRecordDB.get(query.dstAreaId);
    switch (record){
    case (True, *):
      delay = random short delay;
    case (False, True):
      delay = random long delay;
    case (False, False):
      delay = random medium delay;
    }
    scheduleForwarding(query, delay);
  }

B) When a reply message "reply" arrives:
  cancelQueryExipiration(reply.queryId);
  if ((I'm a requesting node || I'm on replying route)
      && unrepliedQueries.isExist(reply.queryId)) {
    transmissionRecordDB.update(query.dstAreaId, r_s,
        True);
    transmissionRecordDB.scheduleExpiration(
        query.dstAreaId, r_s, movingSpeed);
    // After the expiration, go to process D.

    unrepliedQueries.remove(reply.queryId);
  }
  if (I'm on the replying route) {
    transferReply();
  }

C) When a query message "query" is expired
  transmissionRecordDB.update(query.dstAreaId, r_f,
      True);
  transmissionRecordDB.scheduleExpiration(
      query.dstAreaId, r_f, movingSpeed);
  //When the record expires the time, go to process E.

D) When a r_s for dstAreaId is expired
  transmissionRecordDB.update(dstAreaId, r_s, False);

E) When a r_f for dstAreaId is expired
  transmissionRecordDB.update(dstAreaId, r_f, False);
```

Figure 5: Algorithm of forwarding by using GSTR

4 SIMULATION

We evaluated the performance of the GSTR using JIST/SWANS network simulator [7].

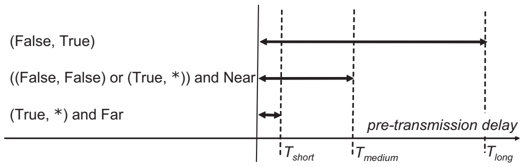

Figure 6: Pre-transmission delay on revised GSTR

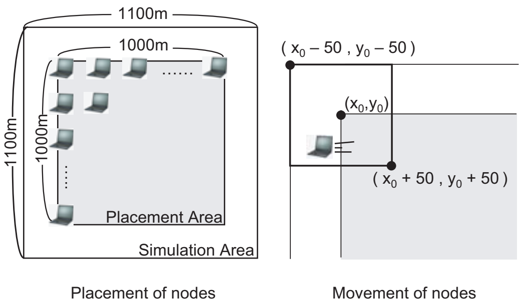

Figure 7: Placement and mobility in this simulation

4.1 SIMULATION MODEL

100–625 mobile hosts move within a square field, 1100 [m] on a side. We call this area $simulation\ area$. All nodes generate query messages for the destination chosen according to a request model defined in 4.1.3. UDP broadcast is used to transmit all messages. We used IEEE802.11DCF for the MAC layer. The bandwidth is 11-Mbit/s. The communication range is 105 [m].

4.1.1 NODE PLACEMENT AND MOBILITY MODELS

In the beginning of each simulation, nodes are initially placed a $placement\ area$, a square area, 1000 [m] on a side, inside the simulation area at even interval as shown in fig.6. Each node moves following random way-point model inside of a square area 100 [m] on a side. The area is placed as the center of it is on the initial position of the node. The moving speed of each mobile host v is chosen randomly from a range 0–2 [m/s] and the pause time is 3 [s].

4.1.2 DATA GENERATION MODEL

The placement area is divided to square sub-areas, 200 [m] a side. Each node has a data item which is related to a sub-area that the node's inital position is included. Each node keeps its own data item. All data items are not replicated.

4.1.3 DATA REQUEST MODEL

All nodes generate query messages for data items following Poisson process with a mean interval of 60 [s]. The packet length of all request message is 128 [bytes] including UDP and IP header. For each query message, a node chooses one of the sub-areas. Thus, data items generated by nodes in the sub-area initially are requested.

4.2 DATA REQUEST AND RESPONSE

Each node forwards request messages to a geocast region by geocast. We used several geocasting schemes for sending query messages in this simulation. Nodes that have requested a data item sends back reply messages containing requested data item to the requesting node. The reply message is simply forwarded by UDP broadcast along the reverse of the requesting route. The size of all reply messages are set to 1000 [bytes] including IP and UDP headers, and the route to the requesting node.

4.3 PERFORMANCE METRICS

Performance metrics used in our evaluation are defined as follows.

- Access Success Raito A_s

$$\mathrm{A}_s = A_c/R_c \tag{2}$$

  R_c(Request count) is the number of all requests and R_c(Answer success count) is the number of successful responses during the whole simulation.

- Traffic per request message T_r

$$\mathrm{T}_r = \frac{T_{forward\_query} + T_{forward\_reply}}{R_c} \tag{3}$$

  Here, $T_{forward\_query}$ and $T_{forward\_reply}$ are the number of packets sent as the query and reply messages, respectively.

- Collisions Rate C_r

$$\mathrm{C}_r = \frac{C_c}{T_{receive\_query} + T_{receive\_reply} + C_c} \tag{4}$$

  Here, C_c is the number of packets which was not received due to collisions and $T_{receive\_query}$ and $T_{receive\_reply}$ are the number of received query and reply messages, respectively.

- Response time per hop

  This is the average response time per hop, time from just after a node sends a query message until it receives the first reply message divided by the number of hops between the requesting node and the replying node.

- Reachability of query messages

  This is the rate of query messages reach a node which maintains requested data item.

- Reachability of reply messages

  This is the rate of reply messages reach nodes which have sent the corresponding query messages.

Table 1: simulation parameters

Parameter	Default value	Range
Simulation time[sec]	10000	
Data size[KB]	1.0	
Number of nodes	100	100 to 625
$v_{\max}$ [m/s]	10	
Pause Time [sec]	3	
Bandwidth [Mbps]	11	
Communication range [m]	105	
Data requests interval [sec]	60	
Number of sub-areas	25	
T_{short} [ms]		10
T_{medium} [ms]		100
T_{long} [ms]		1000
T_w [sec]	2	
T_0 [sec]	30	
k_r	0.5	
$D_{near}[m]$	200	

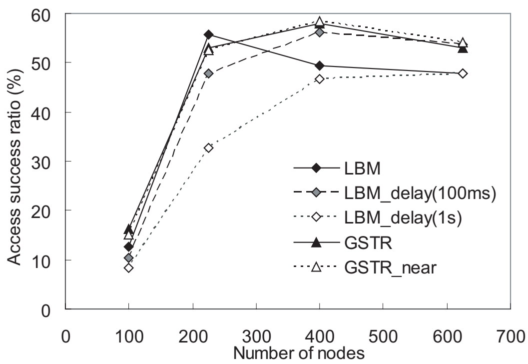

Figure 8: Access success ratio

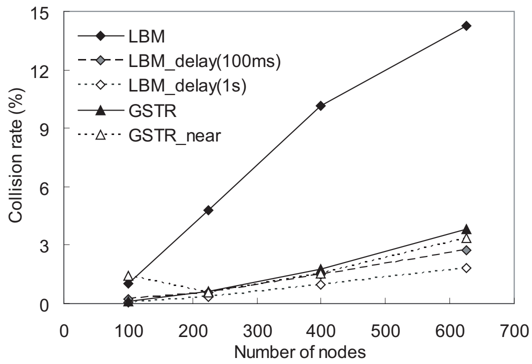

Figure 9: Collision rate

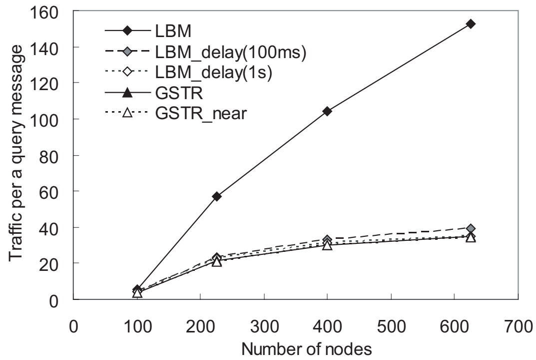

Figure 10: Traffic per a query message

We compared the following four geocasting schemes to examine the effectiveness of GSTR.

- LBM: Query messages are forwarded by simple LBM.
- LBM_delay: Query messages are forwarded by LBM with random delay before transmission of them. We used two different maximum delays, 1 [s] and 100 [ms].
- GSTR: Query messages are forwarded by GSTR. But the pre-transmission delay is not controlled with the distance between a forwarding node and the destination of a query message.
- GSTR_near: Query messages are forwarded by GSTR. Pre-transmission delay is controlled with both successful transmission records and the distance between a forwarding node and the destination of query message.

Table.1 shows simulation parameters.

5 RESULT AND DISCUSSION

Fig.8–13 show the simulation results for various number of nodes. As showing in Fig.8 and Fig.9, when the numbers of nodes is large, A_s of all schemes is small because of the many collisions due to the multiple transmissions of the same message from neighboring nodes.

LBM uses no pre-transmission delay, hence it generates the largest traffic as shown in Fig.10. It achieves slightly higher reachability of query messages than other schemes as shown in Fig.11. However the large mount of redundant messages generates many transmission errors. As a result, the reachability of reply message becomes lower when the number of node is large as shown in Fig.12.

On the other hand, LBM_delay (1s, 100ms), GSTR and GSTR_near drastically decrease C_r. Because these schemes insert delay before the forwarding of query messages, and avoid collisions. As a result of this, A_s of LBM_delay (100ms), GSTR and GSTR_near is higher than A_s of LBM when the number of nodes is large. A_s of LBM_delay (1s) is the lowest of all, because it reduces a large number of collisions with very long pre-transmission delay. As a result, the average response time becomes longer as shown in Fig.13. When the number of nodes is small, the network topology changes fast. Thus the routes used to forward query messages often are unavailable when the reply messages are transmitted. Fig.12 shows reachability of reply messages is low when LBM_delay (1s) is used. On the other hand, when LBM_delay (100ms) is used, the shorter delay before the forwarding of query messages is set to the nodes. Consequently the query messages

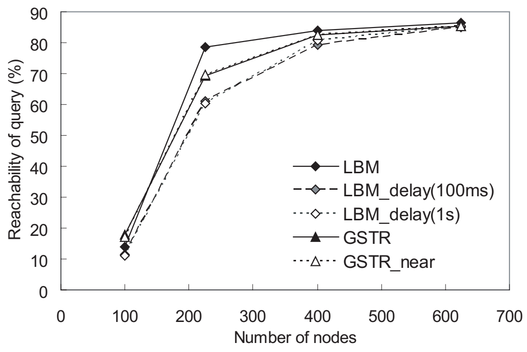

Figure 11: Reachability of query

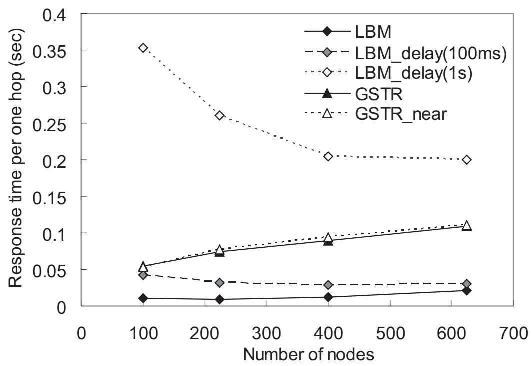

Figure 13: Response time per one hop

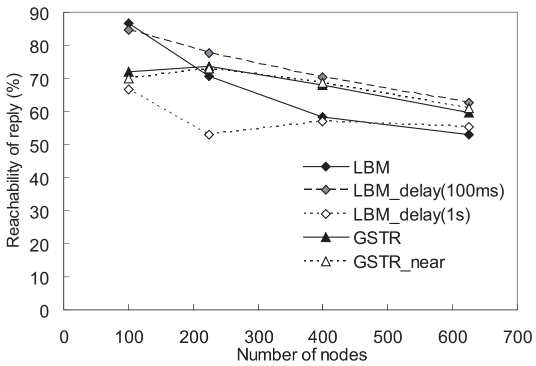

Figure 12: Reachability of reply

are forwarded faster and the reply messages can be sent on the reverse route where the query messages were sent before the route becomes unavailable. Fig.12 shows the reachability of reply messages is the highest of all because LBM_delay (100ms) achieves short response time and low C_r.

T_d and C_r of GSTR and GSTR_near are not the smallest. However A_s of these schemes are the highest. When GSTR is used, the shorter delay before the forwarding of query messages is set to the nodes that have successful transmission record (*True*, *). Thus the query messages are forwarded faster and the reply messages can be sent on the reverse route where the query messages were sent before the route becomes unavailable. As a result, A_s of GSTR and GSTR_near is higher than other schemes. GSTR_near controls the timing of rebroadcast near the destination area of queries. Thus C_r of GSTR_near becomes lower than GSTR. It leads higher A_s of GSTR_near than GSTR.

6 CONCLUSION

In this paper we proposed a scheme which optimizes geocast of queries for location-dependent data using record of successful transmissions. In GSTR, each node collects successful transmission record while transmission of query and reply messages. Each node inserts different delay prior to the rebroadcast on geocast based flooding-based geocast, according to the records.

Simulation results revealed that the proposed scheme reduces due to the multiple transmission of a same query messages and achieves short response time and reduces traffic while it achieves high access success ratio comparing with LBM. Because the proposed scheme avoids congestion of query messages by controlling the delay before the rebroadcast of query messages instead of canceling the transmissions of the query messages, it does not decrease access success ratio. This scheme can be used with cancellation of query messages especially when the density of nodes is high. This enhancement would cope with high access success ratio and short response time even when the node density varies. We will investigate the effect of our scheme when it is used with other optimization schemes in the future.

REFERENCES

[1] Gen Tsuchida, Tomoyuki Okino, Tadanori Mizuno and Susumu Ishihara, "Evaluation of a replication method for data associated with location in mobile ad hoc networks", *in proc. of ICMU 2005*, pp.116-121, 2005.

[2] Gen Tsuchida, Nozomi Suzuki, Mariko Yamanaka, and Susumu Ishihara, "Adaptive replication of location-dependent data in ad hoc networks", *in proc. of INSS2006*, 2006.

[3] Y. B. Ko, N. H. Vaidya, "Flooding-based geocasting protocols for mobile ad hoc networks", *Mobile Networks and Applications*, pp. 471-480, 2002.

[4] S. Ni, Y. Chen, J. Sheu, "the broadcast storm problem in a mobile ad hoc network", *in proc. of ACM/IEEE MOBICOM'99*, pp. 153-167, 1999.

[5] E. Fasolo, R. Furiato, A Zanella, "Smart Broadcast algorithm for inter-vehicular communication", *in proc. of WPMC2005*, pp.1583-1587, 2005.

[6] T. Clausen, P. Jacquet, "Optimized Link State Routing Protocol (OLSR)", *RFC 3626*, 2003.

[7] JIST/SWANS: http://jist.ece.cornell.edu/