A Hybrid Routing Protocol for Mobile Ad Hoc Networks Using Node Encounter Information

Weihua Sun Junya Fukumoto Hirozumi Yamaguchi
Shinji Kusumoto Teruo Higashino
Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita Osaka 565-0871, Japan
{swhua, junya-f, h-yamagu, kusumoto, higashino}@ist.osaka-u.ac.jp

Abstract

We propose a hybrid routing protocol for MANETs where each node maintains potential routes to the nodes which it encountered. Only one route request message is forwarded along the potential route to the destination maintained by the source. Based on this idea, our goal is to reduce the number of control messages by maintaining small amount of information at nodes.

1 Introduction

Mobile ad hoc networks (MANETs) will be one of key infrastructures to make our life more affluent. One practical application of MANETs is extension of coverage area of wireless infrastructure by forming ad hoc networks among neighboring mobile clients. Particularly, such application can be implemented in vehicular ad hoc networks (VANETs). Each vehicle which wishes to access through a base station (BS) to the global network is assisted by VANET to establish a route to BS. MANETs are also useful as a substitution for cellular networks, which will be damaged and disabled in large-scale disaster area. Moreover, requirement for real-time communication between a station and mobile clients in parks, museums or malls is plausible; information terminals are lent at an information center, and people who got the terminals can take benefit of online navigation and location-aware information service over MANETs.

In all the scenarios above, one communication end point is a Base Station (BS in short) which is stationary and may or may not be connected to global networks, and another is one of some Mobile Clients (MCs in short). Also many other Mobile Terminals (MT), which may kindly become constitutes of MANETs, move toward BS or leave from BS. Considering this fact, it is natural (i) to maintain routes between the BS and MCs that are going away from BS and (ii) to provide routes to BS for MCs that are going toward BS by the assistance of MTs, in order to mitigate message overhead of finding routes in on-demand routing protocols.

Motivated by this observation, in this paper, we propose a routing strategy for MANETs and design a corresponding protocol called *Contact-based Hybrid Routing protocol (CHR)*. In CHR, each node records the IDs of nodes which it encountered (called *encounter nodes*). Also, when an encounter node leaves from the node's neighbor group, it finds a relay node that "chains" the encounter node to itself. Since a relay node is also an encounter node, this process is continued and a node chain for each encounter node can be maintained. When a route to a destination node is requested at a source node, the source node specifies the node chain as a potential route if the destination is an encounter node. Only one RREQ message is forwarded along the chain and if a node in the chain finds that its adjacent node is no longer its neighbor, then the node chain maintained by that node is used to fill the gap. If the source has never encountered the destination, the closest node which has the contact entry is found by limited broadcast. As a result, potential routes to MCs leaving from BS can be maintained, and these routes can be used to provide the other MCs approaching BS. We note that obviously this strategy also works well with random-based mobility, where nodes encounter each other in bounded region.

Both a free space region with the random waypoint mobility model and a Manhattan street region with the "evacuation" mobility model were examined in the simulation and the results have shown that CHR could reduce the number of messages while keeping reasonable reachability to destinations, with small amount of information at nodes.

2 Related Work and Contribution

In order to node mobility had been considered harmful for stability of networks, but recently it is regarded as useful for efficient delivery/collection of data. Moreover, to reduce the number of RREQ messages, a lot of research efforts have been dedicated to position-based routing protocols on MANETs (see Refs. [5, 7] for surveys). Compared with the existing routing protocols and message delivery protocols that are aware of node mobility, we do *NOT* assume any knowledge about mobility and node positions, since such assumption will make the protocol lose generality.

The hybrid routing approaches have been proposed to mitigate the overhead of proactive route maintenance. Zone Routing Protocol (ZRP) [8] is an well-known hybrid protocol where the proactive method is used within a routing zone. Most recently, Ref. [4] presented an interesting approach called the orthogonal rendezvous routing protocol (ORRP). In this routing scheme, each node maintains the routing entries of nodes along orthogonal lines, and an RREQ message is forwarded along the lines until it finds an intersection with the line from the destination node. This method assumes directional antennas at each node to determine the orthogonal lines. The CHR protocol is a hybrid routing protocol, but has different goals from the existing ones. For example, ZRP is designed based on the observation where access demands for nearby nodes occur with higher probability, which is reasonable in relatively static networks or large-scale networks. However, if nodes always move like in a city, member nodes may be replaced frequently and such access locality may not be satisfied.

As far as we know, MAID [3] is the first and only one which exploits contact information at each node to determine the direction of message delivery. Unlike the MAID protocol, each node of CHR maintains "node lists" that chain itself and encounter nodes. This has the following two advantages; (i) the node list at a source can be used to specify a route to a destination if the source had encountered the destination, and (ii) the node lists at the other nodes can also be used to fill the gap between two subsequent nodes in the route specified by the source. These features help to increase the possibility to discover a route by only one RREQ message. Even though MAID can determine the direction of messages by traversing newer contact information, the messages may get lost if the nodes which have the contact information had moved in different directions.

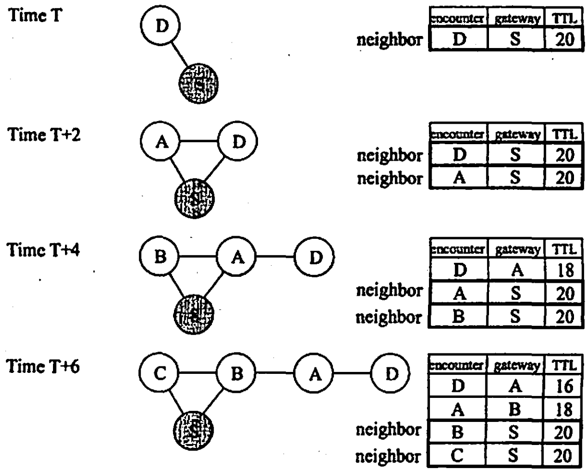
In summary, CHR is an original and effective approach based on a new idea that maintaining routes to encounter nodes will be helpful to reduce routing overhead.

3 Protocol Design

3.1 Protocol Operation Overview

Each node in CHR (say node i) examines link connectivity with it neighbors by periodical beacon messages. Also node i has a table called a *contact table*, which consists of *contact entries*. If node j has/had a link with node i (*i.e.* node j is/was a neighbor), node j is said to be an *encounter node* of node i (and vice versa). A contact entry is maintained for each encounter node of i. We assume that each link is bi-directional. A contact entry consists of an encounter node field, a *gateway node* field and a TTL field;

$$\langle encounter\ node,\ gateway\ node,\ TTL\rangle$$



Time T

	encounter	gateway	TTL
neighbor	D	S	20

Time T+2

	encounter	gateway	TTL
neighbor	D	S	20
neighbor	A	S	20

Time T+4

	encounter	gateway	TTL
	D	A	18
neighbor	A	S	20
neighbor	B	S	20

Time T+6

	encounter	gateway	TTL
	D	A	16
	A	B	18
neighbor	B	S	20
neighbor	C	S	20

Figure 1. Updating Contact Table of Node S

A contact entry $\langle j, k, t\rangle$ of node i means that if node i wishes to find a route to encounter node j, gateway node k is responsible for finding a sub-route from node k to node j and t is the residual lifetime of the entry. If node i detects link connectivity to a neighboring node j and if a contact entry for node j does not exist, node i creates a contact entry $\langle j, i, TTL_{init}\rangle$ where TTL_{init} is an initial lifetime. If a contact entry $\langle j, k, t\rangle$ exists (k may be i), it is initialized to $\langle j, i, TTL_{init}\rangle$. The entry $\langle j, i, TTL_{init}\rangle$ at node i means that node j is the direct neighbor of node i. Here we let each beacon message that examines link connectivity include the list of the transmitter's neighbors. Once node i detects break of the connectivity with neighboring node j, then node i finds a neighboring node k which has connectivity with node j as well as i. The information to find such a node is obtained by beacon messages from the neighbors. If such a node is found, node i updates the contact entry $\langle j, i, t\rangle$ to $\langle j, k, t\rangle$. We assume that each node sends a beacon message for every Δt units of time. For the existing contact entry $\langle j, k, t\rangle$ where $k \neq i$, its TTL field value t is decreased by Δt after each examination of connectivity.

Fig. 1 shows an example where we assume that connectivity of nodes are examined for every two units of time. Node S has one neighbor D at time T, and at time $T + 2$ it has two neighbors A and D. At time $T + 4$, node S detects break of the link connectivity to node D. Thus it designates node A which maintains the link connectivity to node D and node S as the gateway to reach node D. Similarly, at time $T + 6$ node S adds a contact entry for encounter node C, designates node B as the gateway to node A, and updates the TTL fields of the entries.

Then the basic operation of route discovery from node S to node D is described below. If node S has a contact entry for node D, it builds a node sequence $[n_0, n_1, ..., n_w]$ according to its own table such that n_0 and n_w are nodes S and D respectively and n_i is the gateway to node n_{i+1} $(0 \leq i \leq w - 1)$. This is a route to the destination node D

scheduled by node S. Then node S removes itself from the sequence and sends a route request (RREQ) message that includes this node sequence $[n_1, ..., n_w]$ to the neighboring node n_1. We can generalize the RREQ forwarding process as follows. Let us suppose that node n_i receives an RREQ message including a node sequence $[n_i, n_{i+1}, ..., n_w]$. Actually node n_{i+1} *was* a neighbor of n_i when node S designated n_i as a gateway to node n_{i+1}, however at this moment node n_{i+1} may no longer be a neighbor of n_i. Therefore, in order to find a route to n_{i+1}, node n_i utilizes its own contact table, and creates a node sequence $[m_0, m_1, ..., m_{z-1}, m_z]$ such that m_0 and m_z are n_i and n_{i+1} respectively and m_i is the gateway node to m_{i+1} $(0 \leq i \leq z-1)$. This is a route from n_i to n_{i+1} scheduled by n_i. Node n_i substitutes this node sequence for the sub-sequence $[n_i, n_{i+1}]$ of $[n_i, n_{i+1}, ..., n_w]$. As a result, a new node sequence $[n_i, m_1, ..., m_{z-1}, n_{i+1}, ..., n_w]$, which represents a route from node n_i to node n_w partially complemented by node n_i, is obtained. Node n_i removes itself from the sequence and send the sequence $[m_1, ..., m_{z-1}, n_{i+1}, ..., n_{w-1}, n_w]$ to node m_1. By repeating the same procedure until the RREQ message reaches node n_w (*i.e.* node D), the RREQ message can obtain a candidate route from S to D.

In forwarding an RREQ message, the traversed route is recorded and a route reply (RREP) message is sent back to S along the reverse route. A route is approved when node S receives RREP(s)[1] successfully.

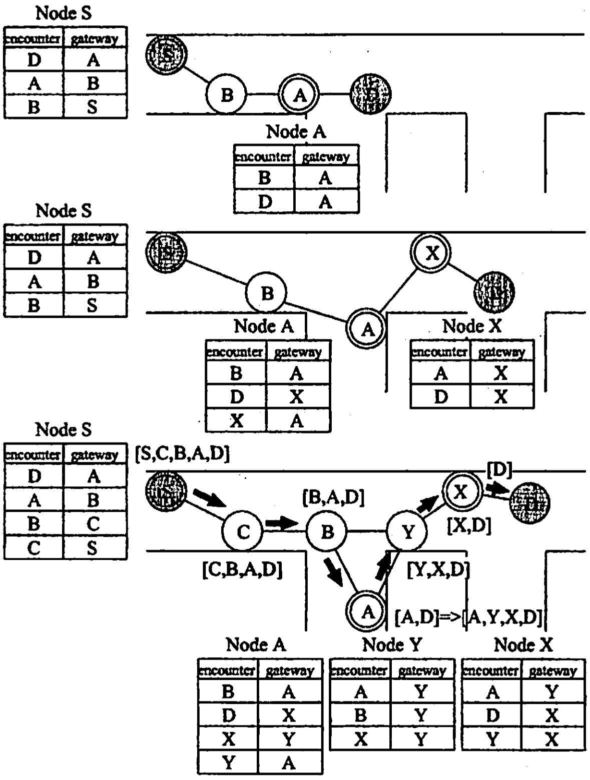


Figure 2. RREQ Message Propagation Using Contact Tables

Fig. 2 exemplifies the route discovery process. For simplicity of drawing, we omitted the TTL fields of contact entries in all the sub-figures. Also nodes represented as double circles are the nodes which encountered the destination node D. Therefore the other nodes represented as single circles have never encountered the destination node D. We assume that the top sub-figure shows a snapshot of the earliest situation, and the bottom shows the latest one. The middle sub-figure shows the situation that node X becomes a gateway to node D at node A because the connection between nodes A and D was broken. We focus on the situation in the last sub-figure; we assume that node S requires a route to node D. Then node S constructs a node sequence $[S, C, B, A, D]$ according to its contact table, removes itself from the head of the sequence and includes the consequent sequence in the RREQ message sent to node C. Node C receives the message and forwards it to node B because node B is its direct neighbor. Node B does the same procedure and then node A receives the RREQ message. During these procedures, S, C and B are removed from the sequence. The RREQ message arriving at node A includes the node sequence $[A, D]$. Here, node D is no longer a neighbor of node A. Therefore node A constructs a node sequence $[A, Y, X, D]$ according to its contact table, and substitutes this sequence for the subsequence $[A, D]$ of the received sequence. As a result, $[A, Y, X, D]$ is obtained. Node A removes itself from the sequence and forwards the RREQ message that includes the sequence $[Y, X, D]$ to node Y.

3.2 Design Consideration

Complementary On-demand Search: If a node needs to establish a route to a destination node which is not an encounter node, an RREQ message is broadcast with limited scope to reach a node that has a contact entry for the destination node. The expanding ring search is one well-known technique that is utilized in many on-demand routing protocols like DSR [6]. We conduct the expanding ring search, and once such a node is found, then the RREQ message is forwarded according to the route discovery procedure explained above. We note that in this case, node D may receive multiple RREQs since more than one node that encountered the destination may be found in the expanding ring search. Thus multiple RREPs are returned through different routes, and node S may select the best one.

Loop Detection and Avoidance: An RREQ message is eventually delivered to the destination using valid contact tables. However, there is a case where a single RREQ message visits a node more than once. Basically in any node sequence generated at a source node, a node does not appear more than once. This is because the contact table contains only one contact entry for each encounter node. However,

[1] Multiple RREP messages through different routes may arrive at node S. We will explain why this happens in Section 3.2.

when a node in the node sequence substitutes its own node sequence to reach the next node, a loop may be generated. The following two cases can be considered.

1. In a node sequence $[n_i, n_{i+1}, ..., n_w]$ at node n_i, the substituted sub-sequence $[n_i, m_1, ..., m_{z-1}, n_{i+1}]$ for $[n_i, n_{i+1}]$ contains node n_v $(i+2 \le v \le w)$.
2. In a node sequence $[n_i, n_{i+1}, ..., n_w]$ at node n_i, the substituted sub-sequence $[n_i, m_1, ..., m_{z-1}, n_{i+1}]$ for $[n_i, n_{i+1}]$ contains node x, which formally forwarded the same RREQ message.

Even though in both cases RREQ messages are delivered to the destination successfully, we may be able to remove this inefficiency as follows. In case 1, node n_i can simply cut the sub-sequence $[n_v, ..., n_{v-1}]$ from the obtained sequence if such a loop appears. In case 2, this loop cannot be removed during RREQ message propagation since the RREQ message has already been forwarded along the loop when the loop is detected. Thus when sending an RREP message back to the source, the destination node can cut the loop from the recorded sequence which the RREQ message has traversed.

Here one may worry about infinite substitution. Since any node sequence generated at any node has finite length, then we can guarantee that the substitution for different sub-sequences eventually ends. Thus we only need to care about the case where the substitution for the same sub-sequence occurs forever. Such situation is modeled as follows. Let us assume that for a node sequence $[n_i, n_{i+1}, ..., n_w]$ in an RREQ message received by node n_i, $[n_i, m_1, ..., m_{z-1}, n_{i+1}]$ is substituted for the sub-sequence $[n_i, n_{i+1}]$ and the new sequence $[n_i, m_1, ..., m_{z-1}, n_{i+1}, ..., n_w]$ is obtained. After being forwarded by several nodes, the RREQ message which includes $[m_j, ..., m_{z-1}, n_{i+1}, ..., n_w]$ is received by m_j. Here we also assume that $[m_j, h_1, ..., h_{s-1}, m_{j+1}]$ is substituted for the sub-sequence $[m_j, m_{j+1}]$ and the new sequence $[m_j, h_1, ..., h_{s-1}, m_{j+1}, ..., m_{z-1}, n_{i+1}, ..., n_w]$ is obtained. Let us suppose the case that the substituted sub-sequence $[m_j, h_1, ..., h_{s-1}, m_{j+1}]$ contains $[n_i, n_{i+1}]$, that is, there exists an index l that satisfies $h_l = n_i$ and $h_{l+1} = n_{i+1}$. In this case the sequence $[m_j, h_1, ..., h_{s-1}, m_{j+1}, ..., m_{z-1}, n_{i+1}, ..., n_w]$ is written as

$$[m_j, h_1, ..., n_i, n_{i+1}, ...h_{s-1}, m_{j+1}, ..., m_{z-1}, n_{i+1}, ..., n_w]$$

which contains n_{i+1} twice. Therefore node m_j detects and cuts the loop, and $[m_j, h_1, ..., n_i, n_{i+1}, ..., n_w]$ is obtained. This obtained sequence indicates that the RREQ message will arrive at node n_i again with the sequence $[n_i, n_{i+1}, ..., n_w]$ and the same substitution for the sub-sequence $[n_i, n_{i+1}]$ will be applied.

In the CHR protocol this situation *never* occurs. To prove this fact, according to the definition of node sequence, we derive the necessary condition for the occurrence of this situation. The condition is as follows; there must be a moment when the following states occur together: (i) m_j has designated n_i as the gateway to n_{i+1} after m_{j+1} left from the m_j's neighbor group. This is required for node m_j to compose the node sequence $[m_j, .., n_i, n_{i+1}, .., m_{j+1}]$. (ii) n_i has designated m_j as the gateways to m_{j+1} after n_{i+1} left from the n_i's neighbor group. This is required for node n_i to compose the node sequence $[n_i, .., m_j, m_{j+1}, .., n_{i+1}]$.

However, these two states are obviously exclusive. The reason is as follows. Without loss of generality, we assume that node m_j enters the state (i) first. Then m_j and m_{j+1} must not encounter each other to maintain the sequence $[m_j, .., n_i, n_{i+1}, .., m_{j+1}]$ at node m_j. However, whenever n_i enters the state (ii), m_j and m_{j+1} must encounter with each other and thus m_j leaves the state (i). Therefore, at any time this necessary condition is not satisfied.

Route Optimization: The loop avoidance discussed in the above paragraph can cut off redundant sub-routes. In addition, the reverse route used to deliver an RREP message can be shorten using the knowledge about one-hop neighbors at each node. For example, for the reverse route $[n_w, ..., n_j, ..., n_i, ..., n_0]$ to the source n_0, if node n_j detects that node n_i is a neighbor, the RREP message can be forwarded directly to n_i.

In general, once a route is found, a route needs no more to follow the contact table. Therefore, several route optimization techniques can be considered, however, we refrain from applying these individual techniques.

Route Error and Repair: If one link is broken on an established route, the gap can be filled by a gateway if exists. Otherwise, the route is repaired by local flooding, or a route error (RERR) message is sent to the source to let it find a new route.

Entry Lifetime: The initial TTL value for contact entries should be tuned according to application scenarios, node mobility and so on. Under low speed mobility, the larger value will lead to better probability of route discovery and with high speed and random mobility the value should be small to adapt to topology changes. We examined three initial values to see the performance difference in the experiments described in the next section.

4 Experimental Results

We have evaluated the performance of CHR by network simulator MobiREAL [1] developed by our research group.

The simulation settings are summarized in Table 1. The area of the simulated regions is 500m×500m, and we have examined the performance of CHR in two scenarios; (1) simulation of communications in free space using the RWP model, and (2) simulation of communications for rescuing

Scenario No.	1	2
Application	-	Disaster Relief
Geography	Free Space	Manhattan Street
Field Size	500×500 (m×m)	500×500 (m×m)
Mobility Model	RWP Mobility	Evacuation Mobility
Simulation Time	1,000 (sec.)	1,000 (sec.)
Number of Nodes (at each moment)	300, 500	200, 300
Speed	[1.0, 2.0] (m/s) (randomly determined at every waypoint)	[0.5, 2.0] (m/s) (randomly determined when initializing nodes)
Communication Request Pattern	Each node generates a request with prob. 0.05 every second to a randomly selected node	A request is generated at the refuge every (2 seconds) to a randomly selected node
Initial TTL Value of Contact Entries	50 (sec.)	50 (sec.)
Radio Range	75 (m)	75 (m)
Beacon Period	10 (sec.)	10 (sec.)
PHY & MAC	IEEE802.11	IEEE802.11

Table 1. Simulation Settings

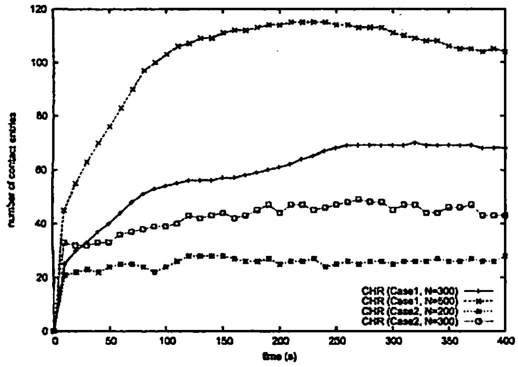


Figure 4. The Average Number of Contact Entries per Node

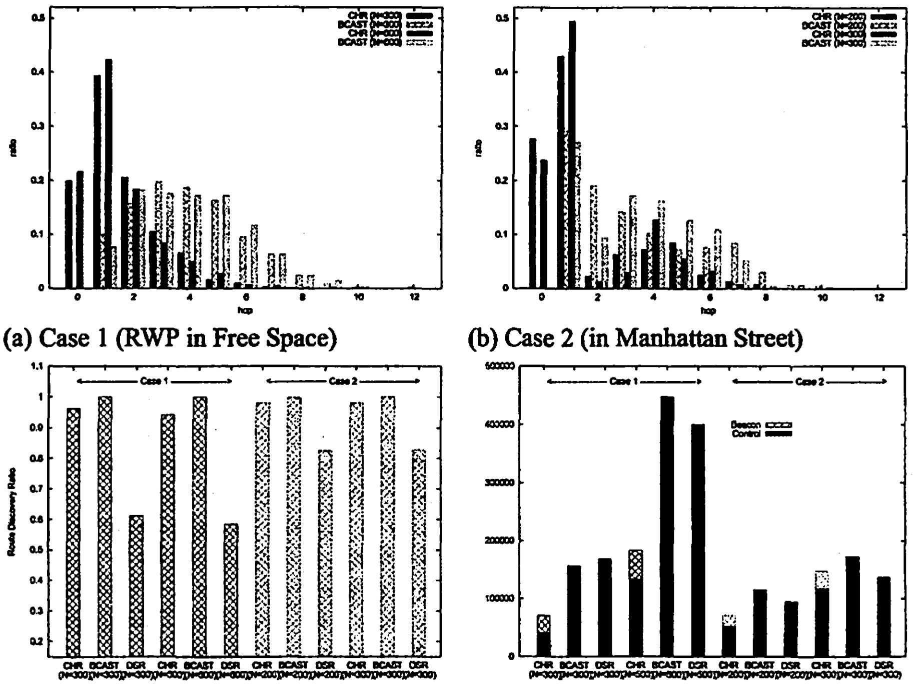
victims in a disaster city section using *Evacuation mobility model*. In the Evacuation mobility model, many persons (evacuees) are swarming and moving toward a place of temporary refuge, and some others (victims) cannot move . A rescue team is located at the refuge, and periodically makes a communication request to a person to know he/she is a victim. These scenarios are referred to as Cases 1 and 2, respectively. In both scenarios, the speed of nodes was set assuming walkers, and the numbers of nodes were set so that the average degrees of nodes can be close between two scenarios (in Case 2, the degree is rather higher due to geography). Usually radio ranges of IEEE802.11 devices are longer than 75m, but considering the fact that too long range will cause too much collision, we set a rather short value. We note that in Case 2, the refuge was placed near the center of the region.

First, in order to see how contact entries are effective to determine routes, for each communication request, we have measured the shortest distance (hops) to find a node which has a contact entry for the destination node. At the same time, we have also measured the shortest distance to the destinations by the broadcast search (denoted by BCAST hereafter) as a benchmark. The experiments are done in two different numbers of nodes in each of Cases 1 and 2. The number of nodes is referred to as N hereafter. The results (distributions) for Cases 1 and 2 are shown in Fig. 3(a) and Fig. 3(b). Each node in CHR could maintain a route (*i.e.* 0 hop) by itself, or find a node which encountered the destination within one hop. Especially, in Case 2, CHR could reduce the ratios of 5-8 hops. This is because the entries for victims, which are 5 to 8 hops away from the refuge were delivered by evacuees to the refuge.

Then we see the route discovery ratio and the total number of packets in Fig. 3(c) and Fig. 3(d), respectively. Cases 1 and 2 are drawn in the same place in each figure. As a benchmark, we have used the DSR implementation of GT-NetS. We have set 50 seconds to the route cache lifetime (this is a rather small value to adapt to mobility). Here the route discovery ratio is the fraction of the route discovery attempts where RREP messages are returned to the sources. The reason why DSR did not perform well is probably cache inconsistency. We may be able to make the cache lifetime shorter, but the number of packets will increase accordingly. BCAST is regarded as a version of DSR without the route cache mechanism. We can see that CHR could achieve enough discovery ratio while maintaining reasonable message overhead even though we take into account the beacon messages which are exchanged to obtain the neighbor information. Especially, these beacon messages in CHR do not concentrate at one time (in the experiments the beacon was transmitted every 10 seconds from each node). On the other hand, in BCAST the route request messages are broadcast at a time and flooded over the network, which gives considerable impact on the transient network load. We note that to make comparison with more optimized broadcasting such as [2] rather than simple broadcast is part of our future work.

Then in Fig. 4 we have measured the average number of contact entries per node. In Case 2, the number of entries is fewer than that in Case 1. This is natural because each node in Case 2 did not meet so many different nodes, instead they met the same nodes frequently due to mobility characteristics. In both cases, the required amount of memory is small enough since each entry only requires space for two node IDs plus a small integer, which is at most a few tens of bytes if we assume the IPv6 addressing scheme.

Finally, Fig. 5 shows the several metrics under different entry lifetime. In Case 2, due to mobility characteristics, the number of entries does not increase as the lifetime becomes larger (Fig. 5(a)). However in Case 2 routes were build over the flows of evacuees, and thus could keep the optimality better than Case 1. For the control packets and the route discovery ratio, the results in Fig. 5(b) and Fig. 5(c) are reasonable. As the lifetime becomes larger, the num-



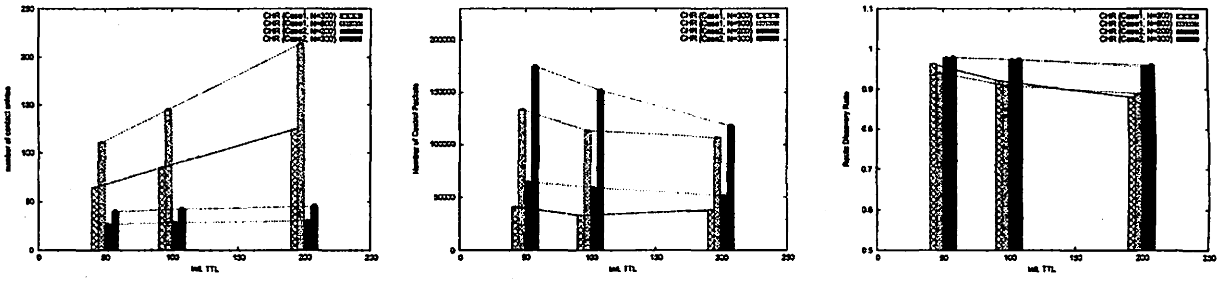
(a) Case 1 (RWP in Free Space)

(b) Case 2 (in Manhattan Street)

(c) Route Discovery Ratio (RWP)

(d) Total Number of Packets (RWP)

Figure 3. Distance to the Closest Contact Entry



(a) Ave. # of Entries

(b) Ctrl. Packets

(c) Ave. Route Discovery Ratio

Figure 5. Several Performance Metrics with Different Lifetime of Contact Entries

ber of packets decreases since entries are well distributed in Case 1, while in Case 2 they are well-delivered using small amount of messages.

5 Conclusion

We have presented a new routing strategy for MANETs and designed a protocol called Contact-based Hybrid Routing protocol (CHR). We have shown through the experiments that this strategy is effective to distribute the route information without causing message overhead. Our scheme is simple, is easy to implement, requires only small information at each node and does not require any other information except connectivity information.

References

[1] MobiREAL web page.. `http://www.mobireal.net`.

[2] J. Arango, A. Efrat, S. Ramasubramanian, M. Krunz, and S. Pink. Retransmission and backoff strategies for broadcasting in multi-hop wireless networks. In *Proc. of BROADNETS*, 2006.

[3] F. Bai and A. Helmy. Impact of mobility on mobility-assisted information diffusion (MAID) protocols. Technical report, USC, 2005.

[4] B.-N. Cheng, M. Yuksel, and S. Kalyanaraman. Orthogonal rendezvous routing protocol for wireless mesh networks. In *14th Int. Conf on Network Protocols (ICNP2006)*, pages 106–115, 2006.

[5] T. Imielinski and J. Navas. GPS-based geographic addressing, routing, and resource discovery. *Communications of the ACM*, pages 86–92, 1999.

[6] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, 353, 1996.

[7] M. Mauve, J. Widmer, and H. Hartenstein. A survey on position-based routing in mobile ad hoc networks. *IEEE Network Magazine*, 15(6):30–39, 2001.

[8] M. R. Pearlman and Z. J. Haas. Determining the optimal configuration for the zone routing protocol. *IEEE Journal on Selected Areas in Communications*, 17(8), 1999.