Journal of Information Processing Vol. 16 27–37 (June 2008)

Regular Paper

A Contact-based Hybrid Routing Protocol for Mobile Ad Hoc Networks

Weihua Sun,[†1] Junya Fukumoto,[†1]
Hirozumi Yamaguchi,[†1] Shinji Kusumoto[†1]
and Teruo Higashino[†1]

In this paper, we propose a routing protocol for mobile ad hoc networks called *Contact-based Hybrid Routing* (CHR) protocol where each node maintains potential routes to the nodes which it encountered. Only one route request message is forwarded along the potential route maintained by the source to the destination. In forwarding the route request message, if an intermediate node finds that the potential route is broken, the node uses the potential route maintained by itself to the next node. Based on this idea, our goal is to reduce the number of route request messages by maintaining a small amount of information at the nodes. The experimental results in random way point mobility and disaster evacuation mobility have shown that CHR could reduce the number of messages while keeping reasonable accessibility to the destinations.

1. Introduction

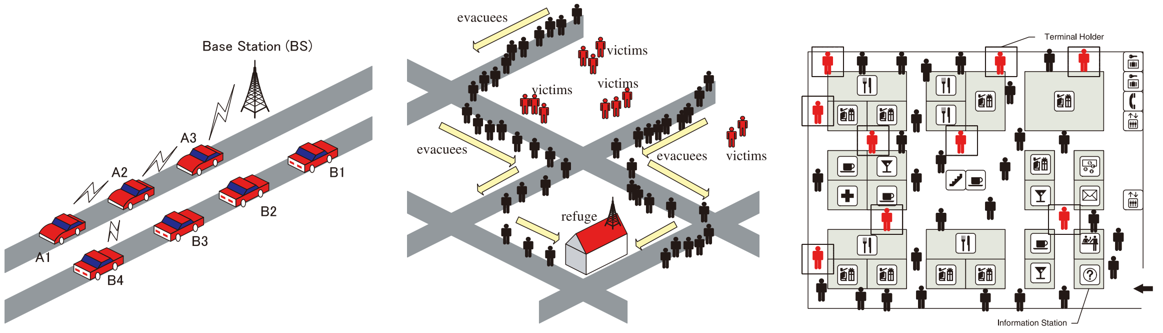
Mobile ad hoc networks (MANETs) will be one of the key infrastructures that will make our life more efficient. One practical application of MANETs is the extension of the coverage area of wireless infrastructure by forming ad hoc networks among neighboring mobile clients. In particularly, such applications can be implemented in vehicular ad hoc networks (VANETs) as shown in **Fig. 1** (a). Each vehicle like $A1$ or $B4$ which wishes to access through a base station (BS) to the global network is assisted by VANETs to establish a route to BS. MANETs are also useful as a substitution for cellular networks, which will be damaged and disabled in a large-scale disaster area. In such a case, a rescue team in the refuge may need to communicate with the cellular phones of disaster victims to know the severity of their situations and to make plans for rescue (Fig. 1 (b)). This is done by the ad hoc facility of brand-new cellular phones which are equipped with wireless LANs. We may compose an ad hoc network among the evacuees to communicate from the refuge to victims who cannot move due to injury or other reasons. Moreover, the requirement for real-time communication between a station and mobile clients in parks, museums or malls becomes possible; information terminals are lent at an information center, and people who have the terminals can take advantage of on-line navigation and location-aware information service by means of MANETs (Fig. 1 (c)).

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 become constituents of MANETs, move towards a BS or away from a BS. In view of this fact, it is natural (i) to maintain routes between the BS and MCs that are going away from the BS and (ii) to provide routes to the BS for MCs that are going toward BS with the assistance of MTs, in order to mitigate message overhead during 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*). The primary goal of our design is to alleviate the overhead caused by Route REQuest (RREQ in short) message flood in reactive routing approaches, without relying on any context information such as location information or special hardware. Also as we target the mobile environment, we would like to avoid the route management cost in proactive routing approaches. To balance these two contradictory requirements, we take a hybrid approach. We simply use connectivity information between neighbors, so we need only periodical messages exchanges between neighbors.

The main idea for establishing a route lies in the following process. Using the connectivity information between neighbors and the same information sent from their (further) neighbors, each node (say node S) knows its two-hop neighbors. Then node S maintains, for each contact node (say node D), a list of nodes which indicates a route to D. This route is obtained as follows; whenever D leaves from the radio range of S, S finds a “relay” node (say node A) which is a common

[†1] Graduate School of Information Science and Technology, Osaka University



(a) Establishing a Route to Base Station over VANET

(b) Victims and Evacuees in a Disaster Area

(c) Terminal Holders on a Floor of a Mall

Fig. 1 Node mobility in the real world.

neighbor of S and D. When A leaves, S similarly finds a relay node B which is a common neighbor of S and A. Continuing this process keeps a route to D from S, like [S, ···, B, A, D]. However, this route may be invalid due to mobility. For example, if B and A have been apart, this route is no longer connected. To cope with this problem, we use a single RREQ message sent from S including route [S, ···, B, A, D]. Whenever a relay node on the route like node B finds that the next node (node A in this case) is no longer a neighbor, it uses its own route information to node A because A had been a neighbor of B and B also keeps track of maintaining a route to A. Therefore, a route is maintained between S and D in a distributed way using only beacon messages between neighbors, and finally this route is found by a single RREQ message. As a result, we can save the number of RREQ messages. 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 a bounded region.

Both a free space region with the random way-point 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 accessibility to destinations, with a small amount of information at nodes.

2. Related Work and Contribution

Node mobility had been considered harmful for the stability of networks, but recently it is regarded as useful for the efficient delivery/collection of data. For example, Ref. 1) has presented an approach where trajectories of messages are scheduled to guarantee delivery of messages in MANET utilizing the knowledge about movement of nodes. Compared with the existing routing protocols and message delivery protocols that are aware of node mobility, we do *NOT* assume any knowledge about mobility, since such an assumption will make the protocol lose generality.

To reduce the number of RREQ messages, a lot of research efforts have been dedicated to position-based routing protocols on MANETs (see Refs. 2), 3) for surveys). Unlike these protocols, we do not require any knowledge of node positions.

The hybrid routing approaches have been proposed to mitigate the overhead of proactive route maintenance. Zone Routing Protocol (ZRP, Ref. 4)) is a well-known hybrid protocol where the proactive method is used within a routing zone. Most recently, Ref. 5) 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 of 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[6)] 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 themselves 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 of discovering 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 have 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

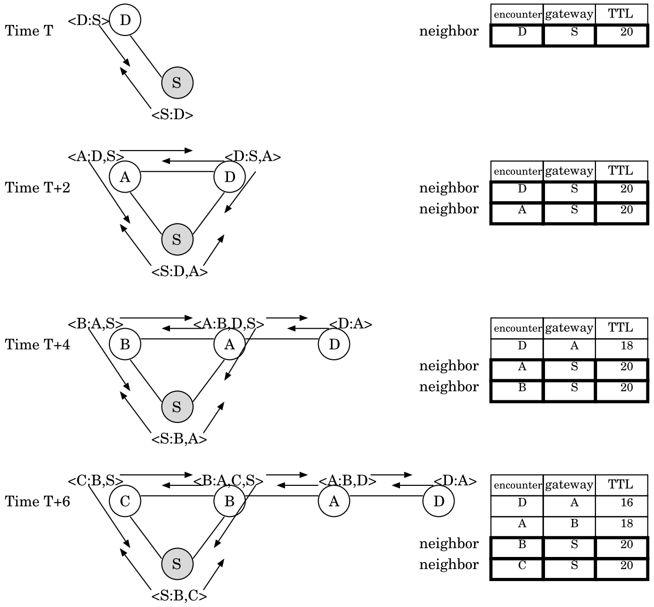
Each node in CHR (say node i) examines the link connectivity with its neighbors by periodical beacon messages. Also node i has a table called a *contact table*, which consists of *contact entries*. If node j has/had a link with node i (i.e., node j is/was a neighbor), node j is said to be an *encounter node* of node i. 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.

$< encounter\ node,\ gateway\ node,\ TTL >$

A contact entry $< j, k, t >$ 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 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 $< j, i, TTL_{init} >$ where TTL_{init} is an initial lifetime. If a contact entry $< j, k, t >$ exists (k may be i), it is initialized to $< j, i, TTL_{init} >$. The entry $< j, i, TTL_{init} >$ 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 a break in the connectivity with neighboring node j, then node i finds a neighboring node k which has a connectivity with node j as well as i. The information for finding such a node is obtained by beacon messages from the neighbors. If such a node is found, node i updates the contact entry $< j, i, t >$ to $< j, k, t >$. Information delivered by beacon messages from a node contains the node's own node ID and the set of IDs of the node's neighbor nodes. CHR does not exchange any contact entry between nodes in order to decrease the message overhead. We assume that each node sends a beacon message for every Δt units of time. For the existing contact entry $< j, k, t >$ where $k \neq i$, its TTL field value t is decreased by Δt after each examination of connectivity.

Figure 2 shows an example where we assume that the connectivity of nodes is 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 a break of the link connectivity to node D. Thus it designates node A which maintains the link connectivity to both 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.

Following this, the basic operation of route discovery to find a route from node S to node D is described below. If node S has a contact entry for node



	encounter	gateway	TTL
neighbor	D	S	20

	encounter	gateway	TTL
neighbor	D	S	20
neighbor	A	S	20

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

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

Fig. 2 Updating contact table of node S.

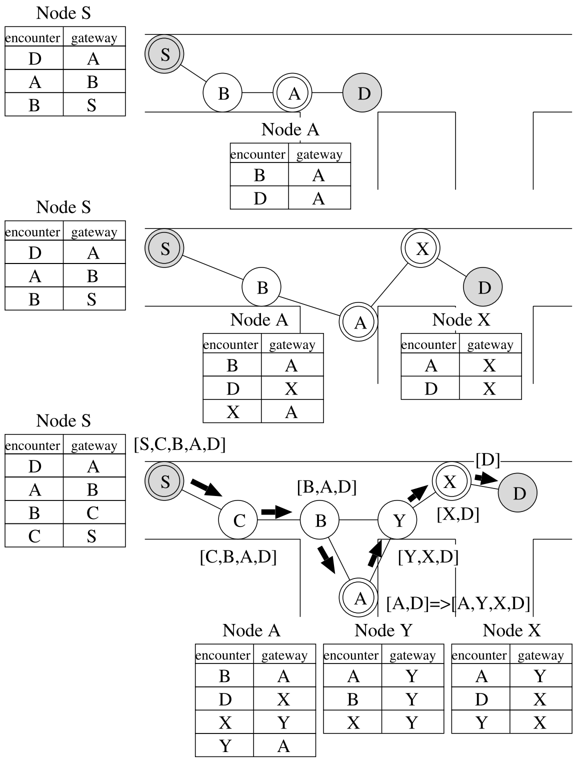
D, it builds a node sequence $[n_0, n_1, \ldots, n_w]$ according to its own contact table such that n_0 and n_w are nodes S and D respectively and n_i is the gateway to node n_{i+1} $(0 \leq i \leq w-1)$. This is a route to the destination node D scheduled by node S. Then node S removes itself from the sequence and sends a route request (RREQ) message that includes this node sequence $[n_1, \ldots, n_w]$ 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 $[n_i, n_{i+1}, \ldots, n_w]$. 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 $[m_0, m_1, \ldots, m_{z-1}, m_z]$ 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 subsequence $[n_i, n_{i+1}]$ contained in $[n_i, n_{i+1}, \ldots, n_w]$. As a result, a new node sequence $[n_i, m_1, \ldots, m_{z-1}, n_{i+1}, \ldots, n_w]$, 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 sends the sequence $[m_1, \ldots, m_{z-1}, n_{i+1}, \ldots, n_{w-1}, n_w]$ 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. We note that we do not refresh TTL of contact entries even when RREP is successfully delivered to the source. In such a case, the initial route may no longer be valid. This is because the initial route may not be the same as the discovered route and in other words, receiving RREP does not always suggest that the contact entries are still valid and effective.

Each node decides a route candidate depending on its entry table. Because there is only one entry for each contact node, we can determine one gateway node for each contact node. Similarly, that gateway node is also a contact node, so one gateway node is found for that gateway node. As a result, only one route candidate is found through the contact table by simple table searching with little overhead.

Figure 3 exemplifies the route discovery process. For simplicity of drawing, we omitted the TTL fields of contact entries. Also nodes represented as double circles are the ones 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 the gateway to node D at node A because the connection between nodes A and D was broken. We focus on the situation in the bottom sub-figure; we assume that node S requires a route to node D.

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



Node S

encounter	gateway
D	A
A	B
B	S

Node A

encounter	gateway
B	A
D	A

Node S

encounter	gateway
D	A
A	B
B	S

Node A

encounter	gateway
B	A
D	X
X	A

Node X

encounter	gateway
A	X
D	X

Node S

encounter	gateway
D	A
A	B
B	C
C	S

Node A

encounter	gateway
B	A
D	X
X	Y
Y	A

Node Y

encounter	gateway
A	Y
B	Y
X	Y

Node X

encounter	gateway
A	Y
D	X
Y	X

Fig. 3 RREQ message propagation using contact tables.

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 find 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[7]. 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 according to some basis such as least hops. In a static MANET, each node only contacts its stationary neighbors, and in this case, CHR has to broadcast RREQs to all over the network.

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, 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 $[n_i, n_{i+1}, \ldots, n_w]$ at node n_i, the substituted subsequence
$[n_i, m_1, \ldots, m_{z-1}, n_{i+1}]$ for $[n_i, n_{i+1}]$ contains node n_v $(i+2 \le v \le w)$.

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

In both cases RREQ messages are delivered to the destination successfully, but we may remove such loops to avoid inefficiency. In Case 1, node n_i can simply cut the sub-sequence $[n_v, \ldots, n_{v-1}]$ 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 be concerned about the case where the substitution for the same sub-sequence occurs forever. Such a situation is modeled as follows. Let us assume that for a node sequence $[n_i, n_{i+1}, \ldots, n_w]$ in an RREQ message received by node n_i, $[n_i, m_1, \ldots, m_{z-1}, n_{i+1}]$ is substituted for the sub-sequence $[n_i, n_{i+1}]$ and the new sequence
$[n_i, m_1, \ldots, m_{z-1}, n_{i+1}, \ldots, n_w]$ is obtained. After being forwarded by several nodes, the RREQ message which includes
$[m_j, \ldots, m_{z-1}, n_{i+1}, \ldots, n_w]$ is received by m_j. Here we also assume that $[m_j, h_1, \ldots, h_{s-1}, m_{j+1}]$ is substituted for the sub-sequence $[m_j, m_{j+1}]$ and the new sequence
$[m_j, h_1, \ldots, h_{s-1}, m_{j+1}, \ldots, m_{z-1}, n_{i+1}, \ldots, n_w]$ is obtained. Let us suppose the case that the substituted sub-sequence $[m_j, h_1, \ldots, h_{s-1}, m_{j+1}]$ 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 $[m_j, h_1, \ldots, h_{s-1}, m_{j+1}, \ldots, m_{z-1}, n_{i+1}, \ldots, n_w]$ is written as $[m_j, h_1, \ldots, n_i, n_{i+1}, \ldots h_{s-1}, m_{j+1}, \ldots,$
$m_{z-1}, n_{i+1}, \ldots, n_w]$ which contains n_{i+1} twice. Therefore node m_j detects and cuts the loop, and $[m_j, h_1, \ldots, n_i, n_{i+1}, \ldots, n_w]$ is obtained. This obtained sequence indicates that the RREQ message will arrive at node n_i again with the sequence $[n_i, n_{i+1}, \ldots, n_w]$ 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 $[m_j, \ldots, n_i, n_{i+1}, \ldots, m_{j+1}]$. (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 $[n_i, \ldots, m_j, m_{j+1}, \ldots, n_{i+1}]$.

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 $[m_j, \ldots, n_i, n_{i+1}, \ldots, m_{j+1}]$ at node m_j. However, whenever n_i enters the state (ii), m_j and m_{j+1} must encounter 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 shortened using the knowledge about one-hop neighbors at each node. For example, for the reverse route $[n_w, \ldots, n_j, \ldots, n_i, \ldots, n_0]$ 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 no longer needs 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 this 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 a 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[8),9)] developed by our research group. The MobiREAL simulator enables

Table 1 Simulation settings.

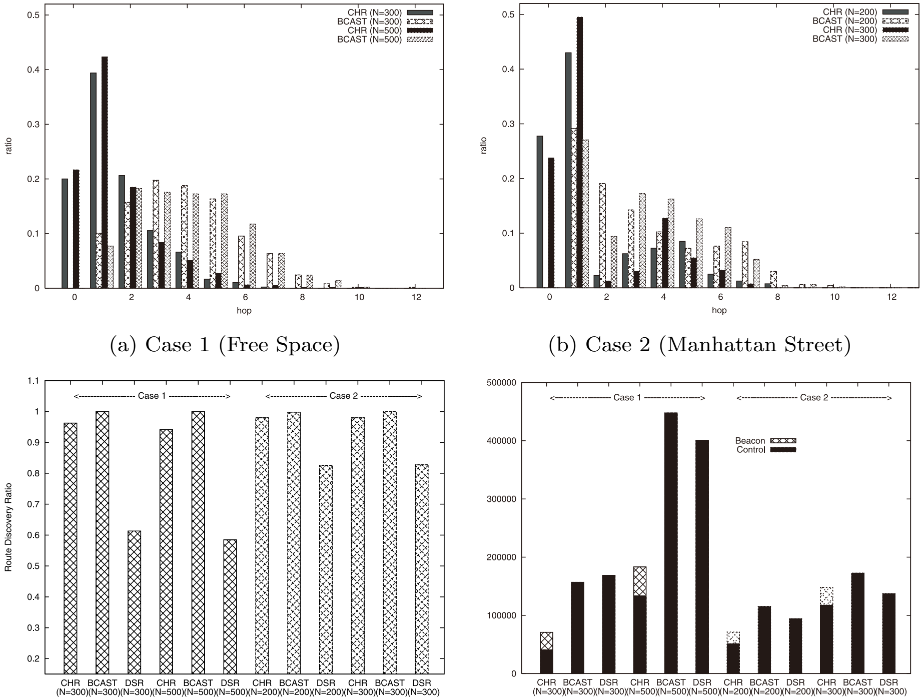
Scenario No.	1	2
Application	-	Disaster Relief (Fig. 1 (b))
Geography	Free Space	Manhattan Street
Field Size	500×500 (m×m)	500×500 (m×m)
Mobility Model	RWP Mobility	Evacuation Mobility
Simulation Time	1,000 (sec.)	1,000 (sec.)
Number of Nodes (at each moment)	300, 500	200, 300
Speed	[1.0, 2.0] (m/s) (randomly determined at every way-point)	[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

accurate simulation of mobile wireless networks by a realistic modeling of the environment (obstacles and movement of nodes). It consists of a network simulator and a behavior simulator that inter-work with each other. The facility of network simulator part relies on GTNetS [10].

The simulation settings are summarized in **Table 1**. The area of the simulated regions is 500 m × 500 m, and we have examined the performance of CHR in two scenarios; (1) simulation of communications in free space using the Random Way Point (RWP in short) model, and (2) simulation of communications for rescuing victims in a disaster city section using the *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 (Fig. 1 (b)). A rescue team is located at the refuge, and periodically makes a communication request to a person to know whether 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 the geography). Usually radio ranges of IEEE802.11 devices are longer than 75 m, but in view of the fact that too long a 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. 4** (a) and Fig. 4 (b) In these figures, the number of nodes is referred to as N. 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. In Case 2, CHR could reduce the ratios of 5-8 hops compared with BCAST. This is because the entries for victims, which are 5 to 8 hops away from the refuge were delivered by evacuees to the refuge. **Table 2** shows "intermediate hops discovered by BCAST", complement broadcast for intermediate nodes is needed in some cases but the ratio is low and the hops are less than 2 hops.



(a) Case 1 (Free Space)

(b) Case 2 (Manhattan Street)

(c) Route Discovery Ratio

(d) Total Number of Packets

Fig. 4 Distance to the closest contact entry.

Table 2 Intermediate hops discovered by BCAST.

Hops	Case 1: 300	Case 1: 500	Case 2: 200	Case 2: 300
1	2.2%	3.4%	2.5%	2.5%
2	0.9%	1.1%	0.6%	0.6%
Other	0.3%	0%	0%	0%

We also did an experiment to test the influence of communication pattern and mobility model on the performance of CHR. We added 2 other scenarios (Cases) (**Table 3**) to achieve this goal. The conclusion is that the mobility model is the main factor. Also, in general, communication patterns do not affect the performance so much, but there is a special communication pattern that may cause low performance. The case is as follows: there are some directional node flows and nodes at the upstream side that find it hard to grab contact information from nodes at the downstream side. The experimental result is shown in **Fig. 5**. The comparison of Cases 1 & 3 does not show any special characteristic. The comparison of Cases 2 & 4 shows that Case4 achieved higher discovery ratios in 2-4 hops. This is because in Case4 the random source node contacted the victims or the nodes around them and as a result, the source node could easily find the contact entries. Notice there are high discovery ratios in 5-6 hops in Case4. This is because the source nodes at the upstream side are hard to get the contact information on the nodes at the downstream side, long distance broadcastings are needed in this case. The comparison of Cases 2 & 3 is similar to the results of 1 & 2.

Table 3 Senarios settings of Case 3 & 4.

Case	Mobility Model	Communication Pattern
3	RWP	fixed source, random dest
4	Evacuation	random source, random dest

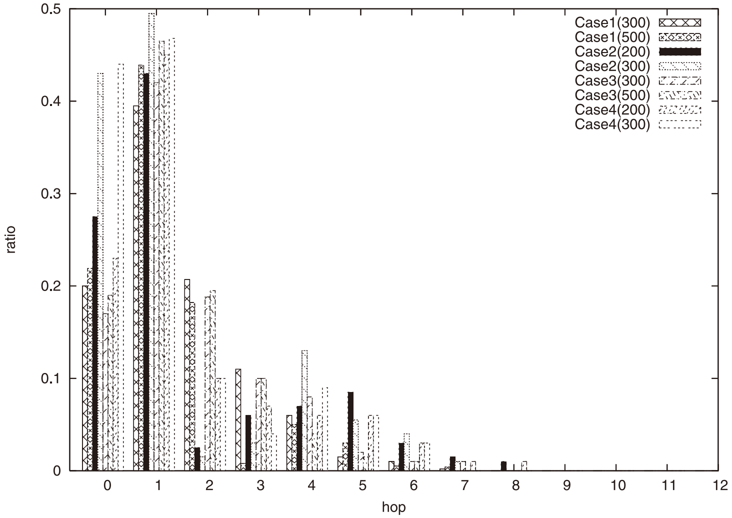


Fig. 5 Influence of communication pattern and mobility model.

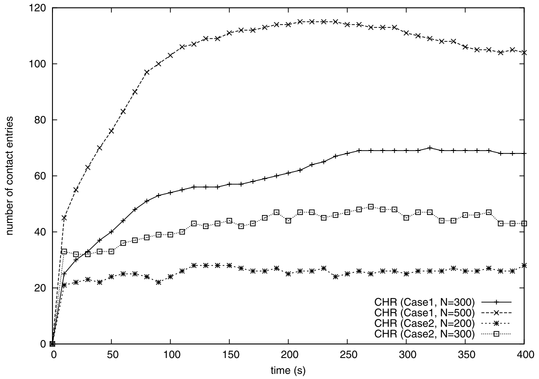
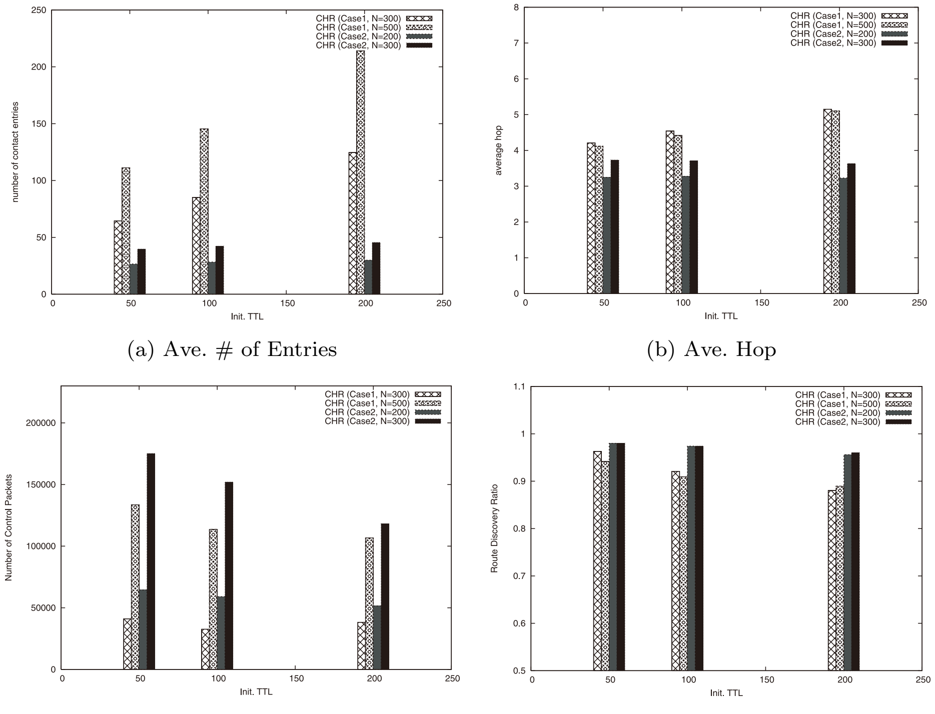


Fig. 6 The average number of contact entries per node.

Following this we see the route discovery ratio and the total number of packets in Fig. 4 (c) and Fig. 4 (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 GTNetS. We have set 50 seconds to the route cache lifetime for DSR (This parameter is not useful in CHR and BCAST). 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. In particular, 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 one time and flooded over the network, which gives considerable impact on the transient network load. We note that to make a comparison with more optimized broadcasting such as Ref. 11) rather than simple broadcast is part of our future work.



(a) Ave. # of Entries

(b) Ave. Hop

(c) Ctrl. Packets

(d) Ave. Route Discovery Ratio

Fig. 7 Several performance metrics with different lifetime of contact entries.

Then in **Fig. 6** 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. 7** shows the several metrics under a different entry lifetime. In Case 2, due to mobility characteristics, the number of entries does not increase as the lifetime becomes larger (Fig. 7 (a)). Also due to the same mobility characteristics, the average hop does not increase (Fig. 7 (b)). Usually as the lifetime becomes larger, routes become longer because each route diverges from the optimum (the shortest path). However in Case 2 routes were built over the flows of evacuees, and thus could keep the optimality better than in Case 1. For the control packets and the route discovery ratio, the results in Fig. 7 (c) and Fig. 7 (d) are reasonable. As the lifetime becomes larger, the number of packets decreases since entries are well distributed in Case 1, while in Case 2 they are well-delivered using small amount of messages. The characteristics are considered the same when TTL is less than 50 or larger than 200.

5. Conclusion

We have presented a new routing strategy for MANETs and designed a protocol called Contact-based Hybrid Routing protocol (CHR). A node in CHR maintains a potential route to each encounter node, and this potential route is complemented by other potential routes maintained by the intermediate nodes. We have shown through the experiments that this strategy is effective in distributing the route information without causing message overhead. Our scheme is simple, is easy to implement, requires only a small amount of information at each node and does not require any other information except connectivity information.

References

1) Li, Q. and Rus, D.: Sending messages to mobile users in disconnected ad-hoc wireless networks, *Proc. Mobicom*, pp.44–55 (2000).
2) Imielinski, T. and Navas, J.: GPS-based geographic addressing, routing, and resource discovery, *Comm. ACM*, pp.86–92 (1999).
3) Mauve, M., Widmer, J. and Hartenstein, H.: A Survey on Position-Based Routing in Mobile Ad Hoc Networks, *IEEE Network Magazine*, Vol.15, No.6, pp.30–39 (2001).
4) Pearlman, M.R. and Haas, Z.J.: Determining the Optimal Configuration for the Zone Routing Protocol, *IEEE Journal on Selected Areas in Communications*, Vol.17, No.8 (1999).
5) Cheng, B.-N., Yuksel, M. and Kalyanaraman, S.: Orthogonal Rendezvous Routing Protocol for Wireless Mesh Networks, *14th Int. Conf on Network Protocols* (*ICNP2006*), pp.106–115 (2006).
6) Bai, F. and Helmy, A.: Impact of Mobility on Mobility-Assisted Information Diffusion (MAID) Protocols, Technical Report, USC (2005).
7) Johnson, D.B. and Maltz, D.A.: Dynamic Source Routing in Ad Hoc Wireless Networks, *Mobile Computing*, Vol.353 (1996).
8) MobiREAL Web Page. http://www.mobireal.net
9) Maeda, K., Sato, K., Konishi, K., Yamasaki, A., Uchiyama, A., Yamaguchi, H., Yasumoto, K. and Higashino, T.: Getting Urban Pedestrian Flow from Simple Observation: Realistic Mobility Generation in Wireless Network Simulation, *Proc. ACM/IEEE MSWiM2005*, pp.151–158 (2005).
10) Riley, G.F.: The Georgia Tech Network Simulator, *Proc. ACM SIGCOMM Workshop on Models, Methods and Tools for Reproducible Network Research*, pp.5–12 (2003).
11) Arango, J., Efrat, A., Ramasubramanian, S., Krunz, M. and Pink, S.: Retransmission and backoff strategies for broadcasting in multi-hop wireless networks, *Proc. BROADNETS* (2006).

(Received August 23, 2007)
(Accepted March 4, 2008)
(Released June 11, 2008)

Weihua Sun received his B.E., M.E., and D.E. degrees in information and computer sciences from Osaka University in 2003, 2005 and 2008. He is working on routing protocol of MANETs and VANETs.

Junya Fukumoto received his M.E. degree in information and computer sciences from Osaka University in 2007. He worked on routing protocol of MANETs.

Hirozumi Yamaguchi received his B.E., M.E. and Ph.D. degrees in Information and Computer Sciences from Osaka University, Japan. He is currently an Associate Professor at Osaka University. His current research interests include design and implementation of distributed systems and communication protocols, especially protocols and services on mobile ad hoc networks and application layer multicast. Also, he is working on some issues of wireless network simulations. He is a member of IEEE and IEICE.

Shinji Kusumoto received the B.E., M.E., and D.E. degrees in information and computer sciences from Osaka Univiersity in 1988, 1990, and 1993, respectively. He is currently a professor at Osaka University. His research interests include software metrics, software maintenance, and software quality assurance techniques, He is a member of the IEEE, the IEEE Computer Society, IPSJ, IEICE, and JFPUG.

Teruo Higashino received the B.S., M.S., and Ph.D. degrees in information and computer sciences from Osaka University, Japan, in 1979, 1981 and 1984, respectively. He joined the faculty of Osaka University in 1984. Since 2002, he has been a professor in the Graduate School of Information Science and Technology at Osaka University. His current research interests include design and analysis of distributed systems, communication protocol, and mobile computing. Dr. Higashino is a senior member of IEEE, a fellow of IPSJ, and a member of IEICE of Japan.