Fault-tolerant ZHLS Routing Protocol for Wireless Mobile Ad Hoc Network

Satoshi Kamidate, Takashi Katoh, Bhed Bahadur Bista and Toyoo Takata
Faculty of Software and Information Science
Iwate Prefectural University
152-52 Sugo, Takizawa, Iwate-ken, 020-0193
JAPAN

Abstract: A Mobile Ad Hoc Network (MANET) is an infrastructureless and dynamically changing topology wireless network. Due to the movement of nodes in the network, path from a source node to a destination mode frequently breaks. Control packets for reconstructing a new path are flooded in the network increasing network traffic. Furthermore, data packets are either lost or re-transmitted (depending upon the application being used) when the path is broken down. In this paper, we consider and construct a detour path for table driven routing protocols for MANET. When the path is broken and if a detour path exists, the packets are routed via detour nodes. We have simulated our proposal using ZHLS and found that the packets delivery ration is higher in our proposed method than the traditional ZHLS routing protocol.

Key–Words: Detour Path, ZHLS, MANET

1 Introduction

Unlike other mobile networks, such as cellular and IP mobile networks, which have wired backbones and centralized controllers, a mobile ad hoc network [1] neither has a wired backbone nor a centralized controller. The network is self-organizing and is suitable for rapid deployment and extension for campus networks. A node in mobile ad hoc network not only acts as an end node (a host) but also acts as a routing node (a router). Because of the mobility of nodes, network topology changes rapidly also. That is, the route from a source node to a destination node dynamically changes. As a result, finding a route to a destination node with minimum communication overhead (control packets) has been a challenging task for researchers for many years.

There are various type of routing protocols designed for ad hoc networks. They can be divided into two categories: proactive and reactive. A good review of ad-hoc routing protocols can be found in [2].

In proactive routing protocols [3], [4], each node floods its link state packet (LSP) throughout the network. Based on the link state packets of all nodes in the network, each node calculates the shortest path to every other nodes and constructs a routing table. When the topology (link state) changes, for example due to the mobility of nodes, LSPs are flooded to reflect the changes and the routing table is updated. A source node refers to its routing table to send data to a destination node and forwards the data to the next hop node which forwards it to its next hop node and so on until the data reaches the destination node.

In reactive routing protocols [5], [6], a source node, which wants to send data to a destination node, first broadcasts a destination node route discovery packet. When the destination node receives the packet, it sends back a route reply packet via the route it received the route discovery packet. When the source receives the reply packet it sends the data packet via the nodes which are included in the route reply packet. If the path is broken, for example due to the mobility of the nodes in the path, the source node initiates the path discovery process again.

In order to reduce the number of route discovery packets (in reactive protocol) or routing table building LSP packets (in proactive protocol), hierarchical design schemes which are both reactive and proactive such as the Zone Routing Protocol (ZRP) [7] and Zone-Based Hierarchical Link State (ZHLS) routing protocol [8] have been proposed. Since the hybrid protocols employ both proactive and reactive schemes, they are considered to be suitable for large networks.

One of the main problems of the MANET is that, as network topology changes so does the path from one node to another. Thus when packets are in route to a destination node and a node that belongs to the path from source to the destination node moves away, i.e. path is broken, the route is reconstructed all over again. This causes not only data loss but also causes a large number of control packets to be flooded in the network for reconstructing the new route. This consumes the scarce resources of nodes.

In this paper, we propose a methodology to construct a detour path proactively so that when the original path is broken the packets are routed via the detour path without reconstructing a new path all over again. The proposed methodology can be used in any table driven routing protocols, but in this paper we consider ZHLS protocol. We will explain why we chose ZHLS in next section. The simulation results show that the percentage of packets delivered to destination nodes in our proposed method is higher than the traditional ZHLS. Unlike the traditional ZHLS, if a detour route exist, the reconstruction of a path is not required in our proposed method, thus there will be no control packets broadcasted in the network.

The paper is organized as follows. In section 2, we will give a brief description of ZHLS, its drawbacks and why we chose ZHLS to simulate our proposal. In section 3, we will explain the detail of our proposed scheme. In section 4, we will present the simulation results of our proposal. In section 5, we will conclude the paper and give some future directions of our proposal.

2 Zone-based Hierarchical Link State Routing

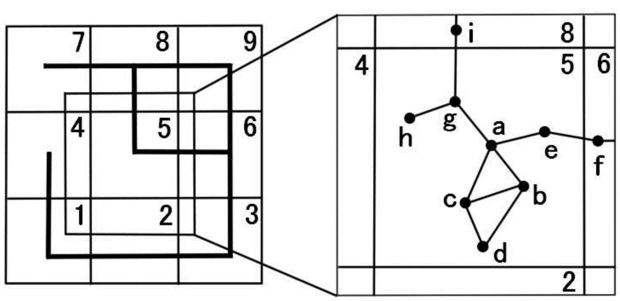

Figure 1: Zone and Node Topology of ZHLS

In ZHLS, an area is divided into zones in which mobile nodes move randomly. A node determines which zone it resides by using GPS or beacon. Each node floods its Link State Packet (LSP). There are two types of LSPs: node LSP for creating routing tables to route packets within a zone and zone LSP for creating inter-zone routing tables to route packets between zones. The Figure 1 shows both node and zone level network topology. Since ZHLS divides an area into zones which can be further divided into sub-zones, it is considered as an appropriate topology for large scale MANET. Furthermore, due to the advancement in GPS technology and reduction of its cost, a GPS receiver can be easily attached to any mobile device and the location of the device can be accurately calculated within a distance of one meter. Considering these two aspects we believe that ZHLS has potential for its use in large MANETs and thus we chose ZHLS as a protocol for simulation of our proposal though our proposal can be applied to any table driven routing protocol.

2.1 Intra-zone Path Construction

Each node within a zone makes a list of its neighbors (nodes to which it has link connection) ID and floods within the zone as Link State Packet (LSP). If a neighbor resides in another zone, it is also included. The detail procedure is as shown below.

1. Each node broadcasts link request.
2. Nodes which receive link request response with their Node ID and their Zone Id.
3. After receiving the responses, the node makes a LSP from the responses.
4. The node then broadcasts its LSP within the zone.

After receiving LSP from the rest of the other nodes in the zone, each node makes a Link State Table from the LSPs and knows the network topology within the zone. The node then uses the Shortest Path Algorithm to create intra-zone routing table. The Table 1(i) and 2 show the Node Link State Table and intra-zone routing table of node *a* in Figure 1 respectively.

Table 1: A part of (i) Node Level Link State Table (LST) of zone 5 and (ii) Zone Level Link State Table (LST) of Figure 1

(i) Node Level LST	
Source	Node LSP
a	b, c, e, g
b	a, c, d
c	a, b, d
d	b, c
e	a, f
...	...

(ii) Zone Level LST	
Source	Zone LSP
1	2, 4
2	1, 3
...	...
5	6, 8
...	...
9	6, 8

2.2 Inter-zone Path Construction

A node that is able to communicate with nodes in another zone is called a gateway node. Each node in a zone knows which zone the gateway node of its residing zone can talk by looking at the LSP of the gateway node. The detail procedure for inter-zone path construction is as shown below.

1. Each node prepares the zone LSP from the node LSPs of the gateway nodes.

Table 2: Intra-zone Routing Table of node *a* in zone 5 in Figure 1

Destination	Next Node
b	b
c	c
d	b
e	e
...	...
h	g
6	e
8	g

2. The gateway nodes flood the zone LSP through out the network.

After receiving zone LSPs from other zones, each node saves them in Zone Link State Table and knows the zone level network topology. Each node then uses the Shortest Path Algorithm to create inter-zone routing table. Table 1(ii) and 3 show the Zone Link State Table and inter-zone routing table for Figure 1 respectively.

Table 3: Inter-zone Routing Table for node *a* in Fig. 1

Destination Zone	Next Zone	Next Node
1	6	e
2	6	e
3	6	e
...	...	...
8	8	g
9	6	e

2.3 Drawback of ZHLS Routing

In ZHLS routing, only one path is constructed to each destination node. When a link in a path is broken, packet delivery is stopped until another path is reconstructed. However, in wireless MANET, due to the movement of nodes or nodes withdrawal from the network, the communication links are broken relatively more frequently and communication stops until another path is reconstructed. This kind of problem is especially more serious for real time applications or streaming data in which retransmission does not take place.

Due to the above problems ZHLS lacks robustness and reliability for MANET. The problem can be solved in some extent by reducing the table update (i.e. broadcast of LSPs) interval but this increases a large number of control packets in the network (network/node resources are expensive in MANET). Frequent table update means frequent execution of the Shortest Path Algorithm which is computationally intensive and not desirable. There are some works, such as [9], which find another path when the present path is broken but they broadcast control packets within a certain range such as TTL to one or so. Our case is different. We do not broadcast packets at all. We prepare detour paths from Link State Table. There is another research which reduces the number of control packets while updating routing table [10].

3 Proposal for Improving Robustness of ZHLS

In ZHLS, if the path to the destination node is broken down due to the broken link or nodes in the path moving away, the packets will not be delivered to the node until a new path is reconstructed. This is due to the fact that only one path from the source to the destination is constructed. In our proposed method, we detour the broken link of the path via peripheral nodes of the path. The node that is used for detour is registered before hand and when the path is broken, packets are forward via the registered node without waiting for the reconstruction of a new path. Suppose in Figure 2(a), node *a* and node *b* are communicating. If the link *a*-*b* is broken the communication is detoured via node *c*. In Figure 2(b), node *a* and *d* are communicating via node *b*. When the node *b* moves away, the communication will continue via node *c* without reconstructing a new path.

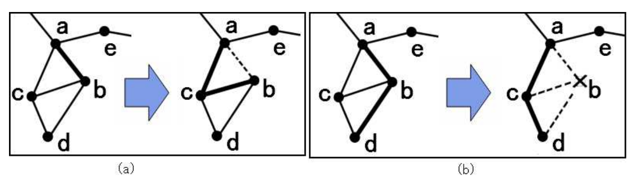

Figure 2: An Example of Detour Path

3.1 Detour Path Construction Method

In our proposed method, a node which is linked to two nodes (directly) of the communication path is used to detour packets when the link between the two nodes is broken. When constructing communication path, neighbor nodes of the nodes on the path are probed and are registered as detour nodes in routing table. In order to do so an extra field in ZHLS routing table is added as shown in Table 4. In the similar manner, detour zones and nodes are registered for inter-zone routing table as shown in Table 5.

3.2 Intra-zone Detour Path Construction

In intra-zone detour path probing, a node first selects a neighboring node as a target node to which a detour

Table 4: Routing Table of Proposed Method for Intra-zone Routing Table of node *a* in Figure 1

Destination Node	Next Node	Detour Node
b	b	c
c	c	b
d	b	c
...	...	...

Table 5: Routing Table of Proposed Method for Inter-zone Routing Table of node *a* in Figure 1

Destination Zone	Next Zone	Next Node	Detour Zone	Detour Node
1	6	e	None	None
...	...	...	...	...
9	6	e	8	g

path is to be constructed. It then checks its Link State Table and finds a node which is linked to itself and the target node and registers the node as the detour node between itself and the target node. The detail procedure for finding a detour path for each target node is as shown below.

1. The probing node (a node which is constructing a detour path) selects one of the neighbor nodes of the target node as the candidate node for the detour path.
2. It then checks whether the candidate node is its neighbor node by checking its NodeLSP.
3. If it is, the candidate node is selected as the detour node of the target node and is registered in the routing table. If it is not then a different neighbor node of the target node is selected as the candidate node and probing is repeated.

The probing node executes the above procedure for all its neighbor nodes and constructs a detour path for each neighbor nodes. Here neighbor nodes of a target node which are not neighbors of the probing node are two hops away. In order to find detour path for those neighbor nodes, they are put in a queue. After probing all the neighbor nodes of the target node, the nodes in the queue are probed as target nodes. In this case also, a node which is linked to both the probing node and the target node is searched in the Link State Table. If it is found it is checked whether it is a target node or not by looking in the routing table. The detail procedure is as shown below.

1. Pop a target node from the queue.
2. Using the Link State Table, select a neighbor node of the target node as a candidate detour node.
3. Check whether the candidate node is neighbor of the probing node.
4. If it is a neighbor node, check whether it is on the path to the target node by looking at the routing table.
5. If it is not, register it as the detour node to the target node. If it is, discard it because it is already on the path.

Repeat the above procedure until the queue is empty. The detour path up to two hops is then constructed. In Figure 2, node *a* and node *b* are part of a path between a source and a destination. The probing node *a* selects neighbor node *b* as a target node and starts probing. Node *a* refers its Link State Table (e.g. Table 1(i)) and obtains the neighbor nodes of node *b*. It finds node *c* which is its neighbor also and registers it as the detour node to node *b*. Moreover, node *a* puts node *d* which is not its neighbor but neighbor of *b* in the queue. Node *a* pops the node *d* from the queue and sets it as a target node and search for the detour node for it. It finds a common neighbors *c* and *b* from the Link State Table. But node *b* is on the path to node *d*, so node *a* excludes node *b* and registers node *c* as the detour node to node *d*. Part of a intra-zone routing table with detour nodes of *a* is shown in Table 4.

3.3 Inter-zone Detour Path Construction

A detour path cannot be constructed for nodes which lie in other zones. This is because ZHLS has hierarchical structures and a node in one zone does not have information of nodes in other zones. Furthermore, there is no mechanism in ZHLS to infer the state of each other's zones. In our proposed method, we use inter-zone routing table and Zone Link State Table to construct a zone level detour path. In order to traverse a zone level path, the packets are in fact forwarded by nodes hop by hop within a zone. Therefore, for zone level path, if the detour path is taken, there is a possibility that the detour path may become much longer than the original path. Thus in our proposed method, we consider the detour path which will have the same number of zone hops as the original path. Moreover, in ZHLS inter-zone routing, packet forwarding occurs by checking the destination node id and its zone id. Therefore for each node, it is sufficient to know the neighbor node and neighbor zone to grasp the path information to the destination node residing in neighbor zone. Here, in order to construct a detour zone path, we consider only neighboring 8 zones. We explain our proposal with Figure 3.

Construction of a zone level detour path is the same as the construction of detour path for two hops away node in node level detour path. Each node selects one of its neighbor zones and obtains its ZoneLSP. Then it selects one of the two hops away zones registered in ZoneLSP and makes it a target

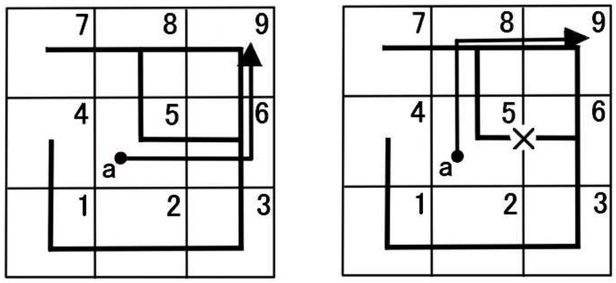

Figure 3: Inter-zone Detour Path Construction

zone. It searches a zone which is linked to both the target zone and the zone it is residing in using Zone Link State Table. By checking the routing table, if the searched zone is not already on the route then it is the detour zone to the target zone and is registered as a detour zone. The detail procedure is as shown below.

1. Obtain the ZoneLSP of the neighbor zone from Zone Link State Table.
2. Select one of the zones registered in ZoneLSP of the neighobor zone as the target zone.
3. Search a zone which is linked to both the target zone and the resident zone (zone where the probing node resides)
4. If such zone is found, make it a detour candidate zone. Using the routing table, check whether the detour candidate zone is next zone to the target zone or not.
5. If it is not, then register the candidate zone as the detour zone to the target zone in the routing table.

We explain the procedure with an example shown in Figure 3 where node *a* in Zone 5 calculates the detour path for zone 9. Node *a* selects its neighbor zone 6 from its Zone Link State Table and obtains ZoneLSP of zone 6 from the Zone Link State Table. From the ZoneLSP of zone 6, it can be found that neighbor zones of zone 6 are zone 3, 5, and 9. Since zone 5 is the zone of node *a* (probing node) it is removed from the search target zone. Lets select zone 9 and make it a target zone. Search a zone which is linked to both zone 5 and the target zone 9 from the Zone Link State Table. Such zones are 6 and 8. Since zone 6 is on next zone to 9, it is removed from the target. Zone 8 is then registered as the detour zone to zone 9. Similarly node *a* searches other zones but there are no zones which are linked to zone 5.

When registering a detour zone in inter-zone routing table, the node also registers the next node to the detour zone as the detour node by looking at its inter-zone routing table. A part of the inter-zone routing table of node *a* in Figure 1 using our proposed method is shown in Table 5.

4 Simulation and Evaluation

In this section, we explain the implementation of the proposed method in a simulation environment and evaluate the packet delivery ratio of the proposed method with respect to the traditional ZHLS.

Simulation was performed using OMNeT++3.2 and AdHocSim1.1 module for ZHLS and proposed method was simulated using C++.

The evaluation is performed by comparing the total packets delivered by our proposed method with the traditional ZHLS. When the movement speed of nodes is changed, we compare the number of packets delivered to the destinatin nodes in our proposed method and the traditional ZHLS.

4.1 Simulation Model

In our simulation, the network is stretched to 840m by 840m and is divided into 9 equal zones. 150 nodes having circular communication range with radius of 100m, are randomly placed in the network. The communication path is updated every 300 seconds. During simulation, nodes in the network move to randomly selected places and stay there until the predetermined time elapses. When the waiting time is over, nodes again randomly select a place and move there. In the simulation, by changing the waiting time, we consider of changing the movement speed of the nodes. The number of packets delivered to the destination nodes is calculated by the number of packets received by the nodes. Packets are sent by 5 nodes to randomly selected nodes in the network.

4.2 Evaluation of Packet Delivery to Destination Nodes

The waiting time of nodes is changed from 30 to 60 with 10 seconds interval. The result for ZHLS is shown in Table 6 and for our proposed method is shown in Table 7. The packet arrival rate with change in waiting time for ZHLS and the proposed method is shown in Figure 4. From the graph, we can show that the proposed method is more robust and tolerant to failure than the traditional ZHLS. During packet forwarding, if the link is broken, control packet in the proposed method will be zero as long as a detour path exists. Thus we did not compare our proposed method with the traditional ZHLS in terms of the control packets. If the detour path does not exists, the control packets generated will be the same as the tradition ZHLS.

Table 6: Number of Packets sent and delivered in ZHLS

Waiting Time	No. of packets sent	No. of packets received	Delivery Ratio
30	2304	1590	69%
40	2326	1473	63%
50	2083	1272	61%
60	2680	1980	73%

Table 7: Number of Packets sent and delivered in our proposed method

Waiting Time	No. of packets sent	No. of packets received	Delivery Ratio
30	2367	2097	88%
40	2710	1999	73%
50	2368	1587	67%
60	2086	1811	86%

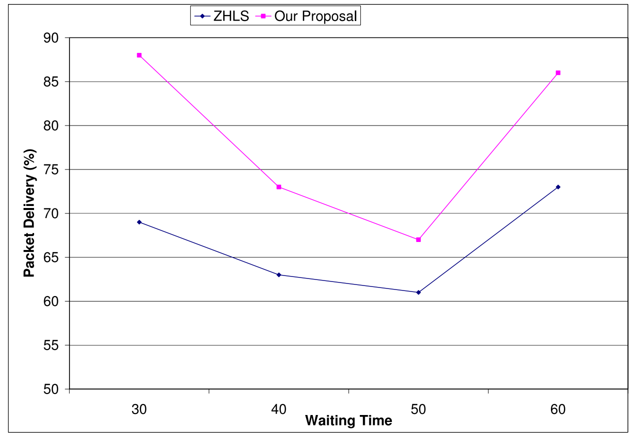

Figure 4: Difference of Pakcet Delivery between ZHLS and our mehtod

5 Conclusions

We have proposed a method for detouring packets when a path from a source node to a destination is broken in MANET. We have applied our method in ZHLS, though it can be used in any table driven routing protocol, and simulated the traditional ZHLS and our proposed method. From the simulation, we found that the packet delivery ratio is higher in our proposed method than the traditional ZHLS. If our proposed method is applied to table driven routing protocols for MANET, it can improve the reliability and robustness of the protocols which is very important for dynamic and infrastructureless network such as MANET. At present, our proposed method considers detour path for only for two hops away nodes. Our future works is to consider a detour path more than two hops away nodes.

References:

[1] C. Toh, *Ad Hoc Mobile Wireless Networks: Protocols and Systems*. Prentice Hall PTR, 2002.

[2] E. Royer and C.-K. Toh, “A review of current routing protocols for ad-hoc mobile wireless networks,” *IEEE Personal Communications Magazine*, pp. 46–55, April 1999.

[3] C. E. Perkins and P. Bhagwat, “Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers,” *Computer Communications Review*, pp. 234–244, Oct. 1994.

[4] S. Murthy and J. Garcia-Luna-Aceves, “An efficient routing protocol for wireless networks,” *ACM Mobile Networks and Applications Journal, Special Issue on Routing in Mobile Communication Networks*, pp. 183–197, Oct. 1996.

[5] D. B. Johnson and D. A. Maltz, “Dynamic source routing in ad-hoc wireless networks,” in *Mobile Computing ed. T. Imielinski and H. Korth*, pp. 153–181, Kluwer Academic Publishers, 1996.

[6] C. E. Perkins and E. Royer, “Ad-hoc on-demand distance vector routing,” in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, (New Orleans LA), pp. 90–100, IEEE, Feb. 1999.

[7] Z. J. Haas, M. R. Pearlman, and P. Samar, “The zone routing protocol (zrp) for ad hoc networks,” *Internet Draft*, July 2002.

[8] M. Joa-Ng and I.-T. Lu, “A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks,” *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 1415–1425, Aug. 1999.

[9] Y. Lin, X. J. Lee, and T. N. Saadawi, “A region-based routing protocol for wireless mobile ad hoc networks,” *IEEE NEtwork Magazine*, vol. 18, pp. 12–17, Jul./Aug. 2004.

[10] T. Hamma, T. Katoh, B. B. Bista, and T. Takata, “An efficient zhls routing protocol for mobile ad hoc networks,” in *17th International Workshop on Database and Expert Systems and Applications (DEXA 2006)*, (Krakow, Poland), pp. 66–70, IEEE Computer Society Press, 4-8 Sept. 2006.