修士論文

アドホックネットワークにおける双方向ルートの競合解決ローカル修復方式

Master's Thesis

Local Repair with Race Resolution for Duplex Route in Ad Hoc Networks

東京大学大学院　新領域創成科学研究科　基盤情報学専攻

Graduate School of Frontier Sciences, The University of Tokyo
Department of Frontier Informatics

36314
張　萌
Meng Zhang

指導教員　若原　恭　教授

2005 年 1 月

Abstract

A Mobile Ad Hoc Network (MANET) is a collection of wireless mobile hosts. Local repair is considered to be an efficient method to repair a route in MANET. However, when more than one link failures occur in a route almost at the same time, there can be a race problem and it may lead to bad network performance especially if the route is duplex. In this paper, we analyze the race problem and propose a new local repair scheme with race resolution for duplex route. We implement it into AODV, which is a typical routing protocol for MANET and demonstrate its effectiveness through some computer simulations.

Acknowledgements

I would like to gratefully acknowledge the enthusiastic supervision of Prof. Yasushi Wakahara during this work. His seriousness on everything influences me at all time. He educated me not only the technologies but also the philosophy for living and being a useful person. I greatly thank the teaching assistant Dr. Fumitaka Nakamura for the technical discussions on my research and for his assistance with all types of technical problems of my computer simulations. I appreciate Associate Prof. Masaya Nakayama for advices and commons in laboratory meetings.

All the Postgraduates of the campus network laboratory are thanked for numerous stimulating discussions, help with experimental setup and general advice; in particular I would like to acknowledge the help of Tomoki Murota for his computers when I need more machine sources for simulations.

Finally, I am also grateful to the staff of the Information Technology Center for their support especially the secretary Yoshizawa and the secretary Kawasaki.

Table of Contents

Figures List

CHAPTER 1 INTRODUCTION

1.1 Motivation

Mobile Ad Hoc Networks are constructed by groups of mobile wireless nodes (ex. laptop computer and mobile phone). In Ad Hoc Network, all the nodes can relay packets like a router, and thus the nodes can have multi-hop communication without any infrastructures such as routers and base stations. Because the nodes are portable and mobile, the communication or packets relay may be cut off due to the battery exhaustion or move beyond their communication range limits. Therefore, routing and route maintenance are big challenging problems in the research of Ad Hoc Networks.

In Ad Hoc Networks there are two kinds of routing protocols, which are table-driven routing protocols and on-demand routing protocols. In table-driven protocols, routing information for all the nodes have to be updated, whenever the network topology changes. Thus, in the case of highly dynamic topology network, the table-driven approach is considered unsuitable [1]. DSR [2] and AODV [3] are some typical on-demand routing protocols that cover this defect of the table-driven approach. In those protocols, a shortest route can be established between the source node (the sender) and the destination node. When the network topology changes fast, reconstruction of the route is frequently necessary. In such a dynamic condition, the method of local repair which repairs a route locally reconstructs a broken route with smaller overhead in a shorter time in comparison with the global repair method [4]. Local repair is applied to many routing protocols such as AODV and ABR [5]. However, because local repairs are usually invisible to the source node, when more than one link failures occur in a route almost at the same time, there can be a race problem especially in the case of repairing a duplex route. The race may lead to bad network performance especially if the route is duplex. In this paper, our purpose is to clarify the race problem and to propose a new local repair scheme with race resolution for duplex route.

1.2 Thesis Outline

In Chapter 2, we introduce present research on routing protocols and route repair method. After introducing the background, we will analyze the race problem of route repairs and reveal their conditions in Chapter 3. In Chapter 4, we propose a new method with race resolution of local repair for duplex route, and also show how to implement this method in AODV in Chapter 5. We did some computer simulations and the details of simulations are described in Chapter 6. The proposed method is evaluated in comparison with some other methods in Chapter 7. Finally the conclusion and future work are given in Chapter 8.

CHAPTER 2 BACKGROUND

2.1 Route Repairs in Routing Protocols

There are two kinds of method to repair a route break in an on-demand routing protocols. The first method is called global repair[3][7][10]. In the global repair, when a link breaks in a route, the source node will be notified of the break and it initiates a new discovery for another route. In the global repair, the source node can always get a shortest route to the destination node. But on the other hand, the source node has to flood a route request message around the network to find the route to the destination at every route discovery. During the repair time, the source node has to suspend its data transmission until the new route is established. The global repair method is adopted in AODV, DSR, etc.

The second method is called local repair[3][5][8]. In the local repair, when a link failure is noticed, it will be repaired in the local area around the failure without notifying the source node. Thus, TTL (Time To Live) value of the repair control message can be small and the route will be repaired rapidly. However, the local repair often leads to a longer alternative route with large probability. Thus the route tends to be longer compared with the global repair. The local repair method is applied to AODV, ABR, etc.

2.2 AODV and Local Repairs in AODV

We take AODV for an example in this paper. AODV provides route discovery function and route repair function as well. It has both global repair function and local repair function as mentioned before.

When a node has a data sending demand, it floods a Route Request (RREQ) message to find a route to the transmission destination. The data sender is called the source node, and the data receiver is called the destination node. Every node that has received the RREQ establishes or updates a reverse route to the source node. When the destination node receives the RREQ, it replies with a Route Reply

(RREP) message. The RREP uses the reverse route to reach the source node.

In AODV, a sequence number, which is a monotonically increasing number, is maintained by each node in order to avoid the reception and processing of the same flooding messages and to designate the different messages of the same name. In a duplex route, every node maintains a Source node Sequence Number (SrcSeqNo) and a Destination node Sequence Number (DstSeqNo). Duplex route is useful when two-way communication such as TCP is necessary. This paper focuses on repair methods of such duplex route. Whenever the Source Node broadcasts an RREQ, the SrcSeqNo is incremented, and other nodes manage the SrcSeqNo when they receive the RREQ. And when the destination node sends an RREP, DstDeqNo is incremented and it will be notified to every node in the new route by the RREP. Thus, each node in the new route has a higher DstDsqNo than those of the nodes in the old route (i.e. those out-of-date broken route). When a route failure happens, the node initiating a route repair also increments the DstDsqNo and puts it into the RREQ.

If the route breaks, the upstream node (i.e. the node nearer to the source between the two ends of the link) of the link first chooses either the global repair method or the local repair method. If the destination is more than MAX_REPAIR_TTL hops away from the upstream node of the link, the upstream node sends a Route Error (RERR) message to the source node to start the global repair. Otherwise, it sends a RREQ with a small value of TTL to the destination to repair the route locally. The value of MAX_REPAIR_TTL is usually set at the value of half the number of hops of the whole route.

The RREQ message for Local Repair has the same format with that for Global Repair as shown in Figure 1. In Figure 1, 'Originator' represents the node that flooded the RREQ message. Figure 2 shows how the Local Repair repairs a route to the destination node but can not repair the reverse route from the destination to the source. The shaded fields in the routing Tables denote the fields created or updated by the local repair. In the new route, node N1 and N2 do not have entries for the source node S and the destination node D still has an incorrect entry for node S in their routing tables. Thus only the one-way route to the destination is repaired.

Type	J	R	G	D	U	Reserved	Hop Count
RREQ ID							
Destination IP Address							
Destination Sequence Number							
Originator IP Address							
Originator Sequence Number							

J, R: reserved for multicast G: Gratuitous RREP flag

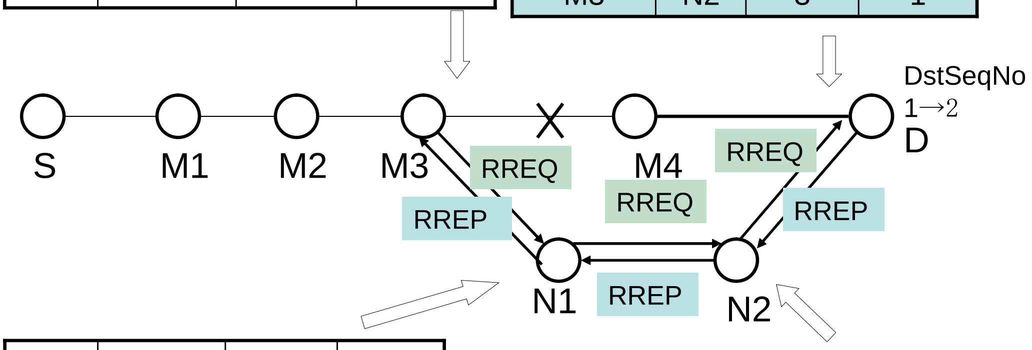
D: Destination only flag (indicates only the destination may respond to this RREQ)

U: Unknown sequence number

Figure 1 Format of RREQ

Dst	Next Hop	Hop Count	SeqNo
D	M4→N1	2→ 3	1→ 2
S	M2	3	1

Dst	Next Hop	Hop Count	SeqNo
S	M4	5	1
M3	N2	3	1



Dst	Next Hop	Hop Count	SeqNo
D	N2	2	2
M3	M3	1	1

Dst	Next Hop	Hop Count	SeqNo
D	D	1	2
M3	N1	2	1

Figure 2 Local Repair of Only Source to Destination Route in AODV

CHAPTER 3 PROBLEMS OF LOCAL REPAIR IN AD HOC NETWORK

3.1 Race among Repairs

Because the local repair is usually autonomous and decentralized, it is difficult to know other links' conditions while a local repair starts. When two route repairs which repair the same route are initiated, race between them happens. This kind of race may have a negative effect on the repair of the route. If a routing protocol has both the global repair and the local repair method (e.g. AODV), then races between global repair and local repairs may also happen.

3.2 Race Happening Conditions

When more than one local repair processes are executed for the same route almost at the same time, race may happen. Figure 3 shows this temporal condition of race between local repairs.

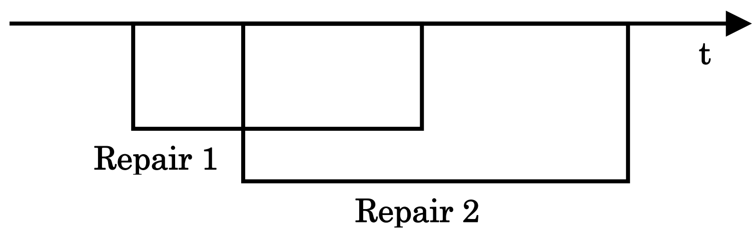


Figure 3 Temporal Condition of Race Between Local Repairs

There are three categories of route failure detection methods. The first is by means of hello packets. In this method, all the nodes broadcast hello packets periodically to their neighbors to notify their connectivity. Once a node becomes unable to receive a hello packet from its neighbor, then the link with the neighbor is taken as broken. The second method is to predict link failures[7][9][10], and each node in the route predicts failure of a link with each of its neighbors by observing the level of radio signals from them. The third method is based on the notice from a node in the route when it failed to transmit a data packet.

In the former two route failure detection methods, when a repair is being processed, other one may be started, when some nodes in the route move fast. In the third method, when the route has lower capacity and the data transmission rate is rather high, packets will be buffered in the queues of the nodes in the route as shown in Figure 4, and thus more than one link may likely to fail and are detected almost at the same time. In addition, if the number of hops of the route is large, more than one link may likely to fail and will also be detected as shown in Figure 5.

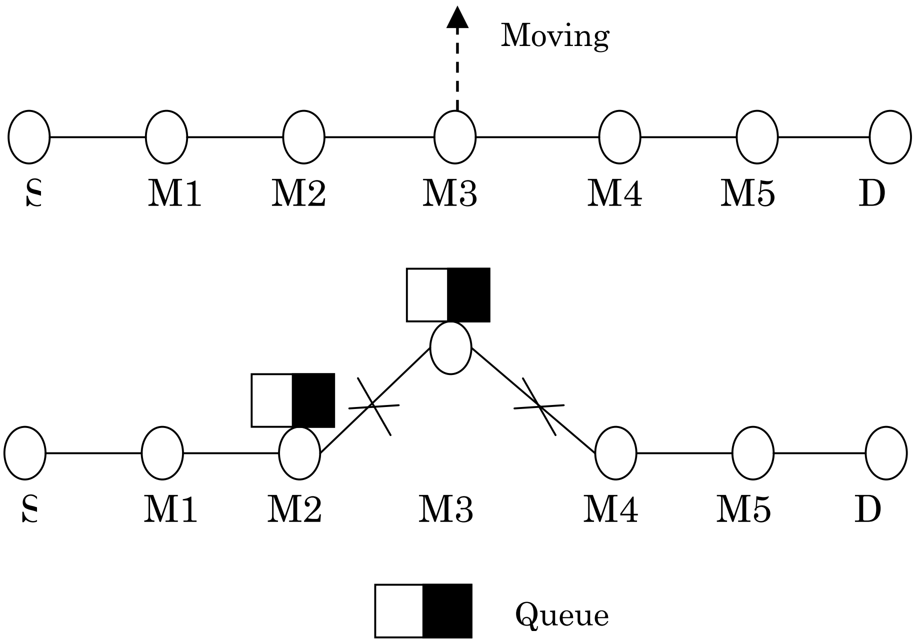


Figure 4 Example 1 of Race: More Than One Local Repairs Occur at Almost the Same Time

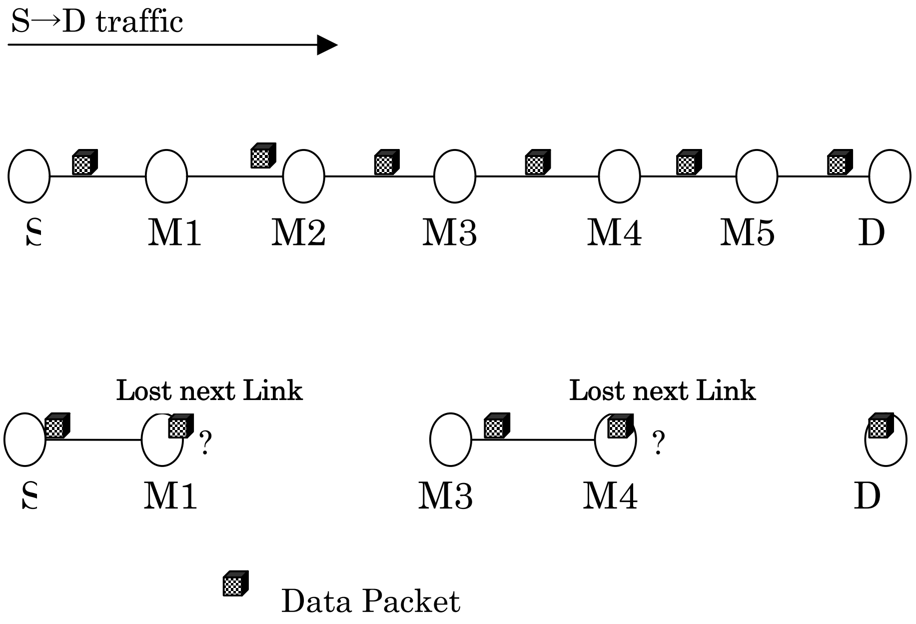
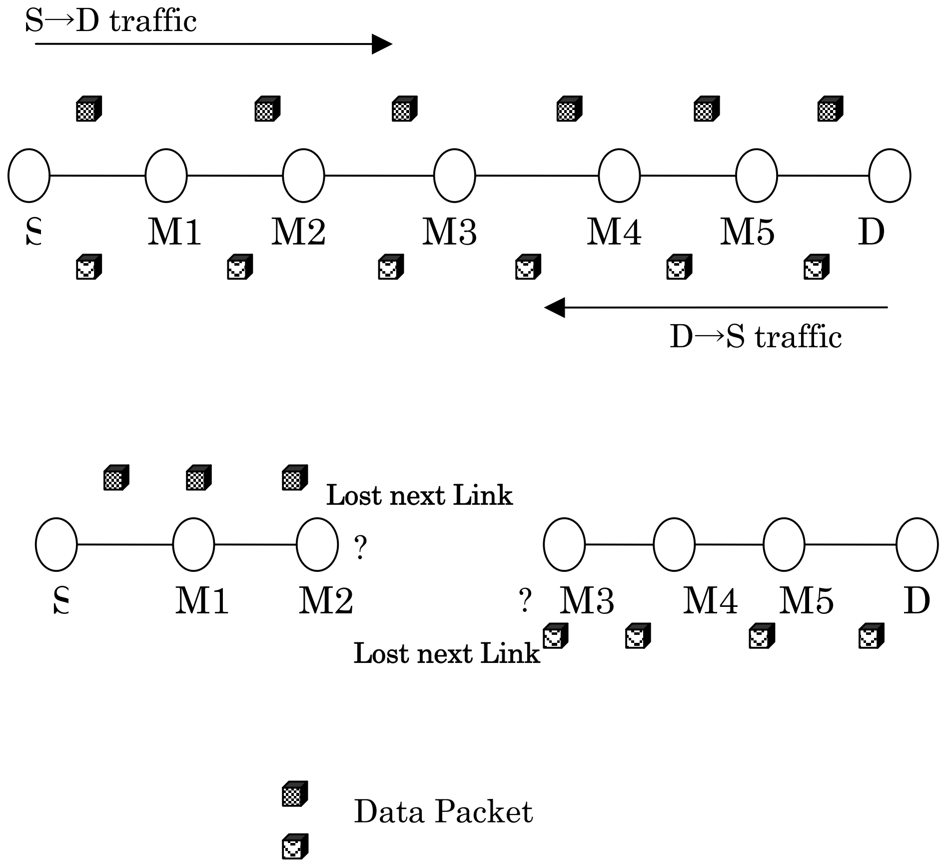


Figure 5　Example 2 of Race: More Than One Local Repair Occur at Almost the Same Time

If two-way traffic flows at the duplex route, even one link break will cause more than one link failure detections may happen. Figure 6 shows the image of this situation.



I

Figure 6 Example 3 of Race: More Than One Link Failure Detections Occur in Case of Two Way Traffic

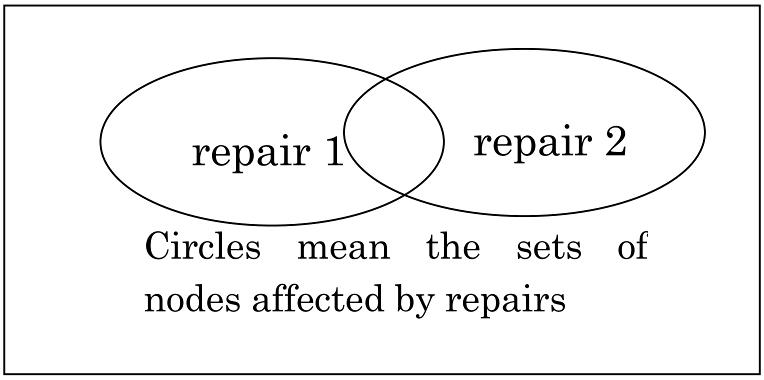


Figure 7 Spatial condition of race between Local Repairs

However, when more than one local repair processes occurred almost in the same time, if they repair, then the race problem does not happen. Figure 7 show the spatial condition of race between local repairs. Taking AODV for instance, RREQ in a Local Repair is flooded to discovery the destination node. Thus, when two local repairs occur in the same route, the nodes near the destination may receive both of their RREQ messages and race happens. In routing protocols with methods of both global repair and local repair, it is also possible that in a route that a global repair and a local repair race for repairing a same route. In that case, global repair can be regarded as a special kind of local repair.

3.3 Problems Caused by Races

3.3.1 Problems in repair simplex routes

If the local repair is used to repair a one-way (i.e. source→destination) route, the races among route repairs usually result in generation of redundant routes. Figure 8 shows the image of the redundant RREQ flooding while race happened in AODV.

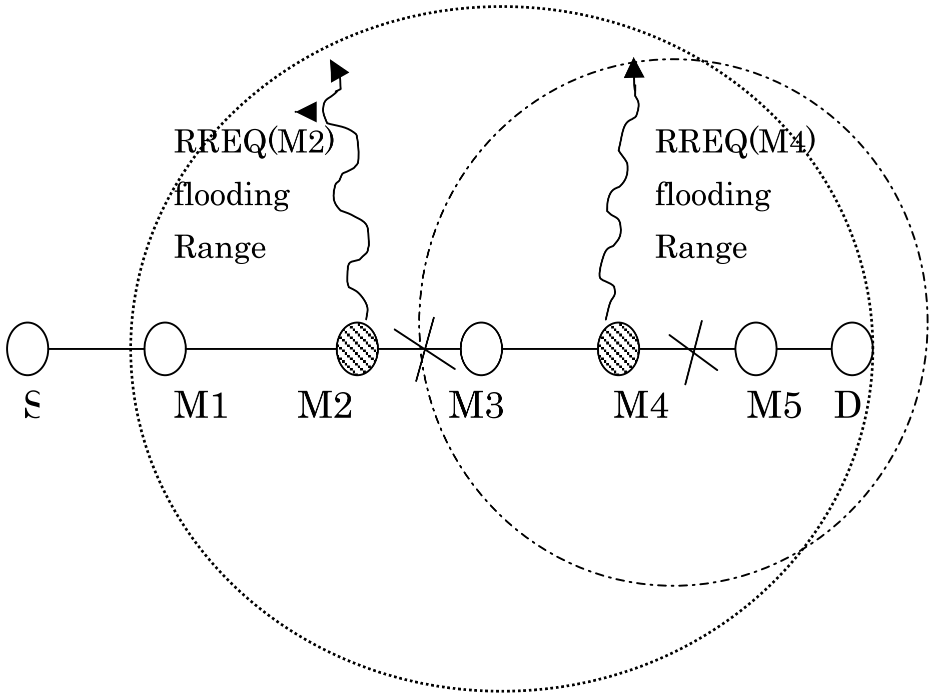


Figure 8　Image of Redundant Overhead Generation When Race Happened for Repairing a Simplex Route

3.3.2 Problems in repair duplex routes

In repair duplex routes, there are two kinds of race may happen. They are the same direction race and the reverse direction race separately which are described as following.

① Same direction race

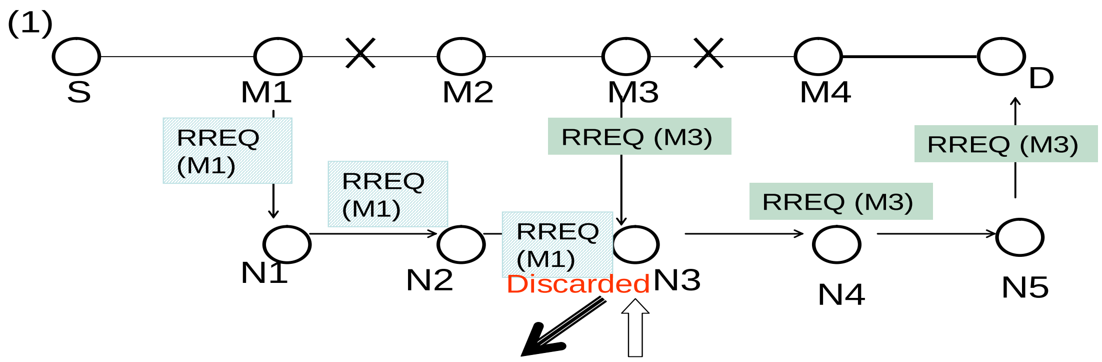
The same direction race is a race that two repairs are trying to repair the same route with the same destination.

In the case of a same direction race with two local repairs, the two repair processes try to repair the same route.

The race may result in the failure of the local repairs and the generation of some redundant useless route as described below where AODV is taken as an example routing protocol.

Figure 9 shows an example of the race between two local repair processes repairing a duplex route in AODV. Node S is the source node and node D is the destination, and between nodes S and D, there is a duplex route S-M1-M2-M3-M4-D. Data traffic from node S to node D uses this route. When links M1-M2 and M3-M4 break in the route, node M1 initiates a local repair by flooding an RREQ (denoted by RREQ(M1)), and almost at the same time node M3 initiates another local repair by flooding an RREQ (denoted by RREQ(M3)). RREQ(M3) firstly reaches node N3 and N3 creates a route entry for node S with node M3 as next hop. Then, RREQ(M3) is forwarded by N3 to other nearby nodes. Shortly later, RREQ(M1) also reaches node N3. However, because there is an entry for node S with the same SrcSeqNo in the routing table of N3, and Hop count of M1 (the number of hops between M1 and S) is not larger than that in the entry, RREQ(M1) received is ignored and discarded. Figure 9(2) shows the result of the race in Figure 9(1). Node D returned an RREP when it received RREQ(M3) and when the RREP arrives at node N3, node N3 forwards it to node M3, referring to the routing table. Because there is no links from M3, the RREP cannot be forwarded to any other nodes, and thus the RREP cannot repair the route, which leads to the failure of the route repair. In due course of time, M1 will notice the repair failure and reports it to node S. Finally, node S initiates a global repair by flooding an RREQ with large TTL. In this case, not only a redundant partial route M3-N3-N4-N5-D is maintained in the network until it is cleared because it is not used during a predetermined time period, but also the data transmission has to be suspended until the global repair succeeds. The packets cashed in M3 are obliged to be discarded because of the local repair failure. Thus the network performance deteriorates due to the race.

(1)



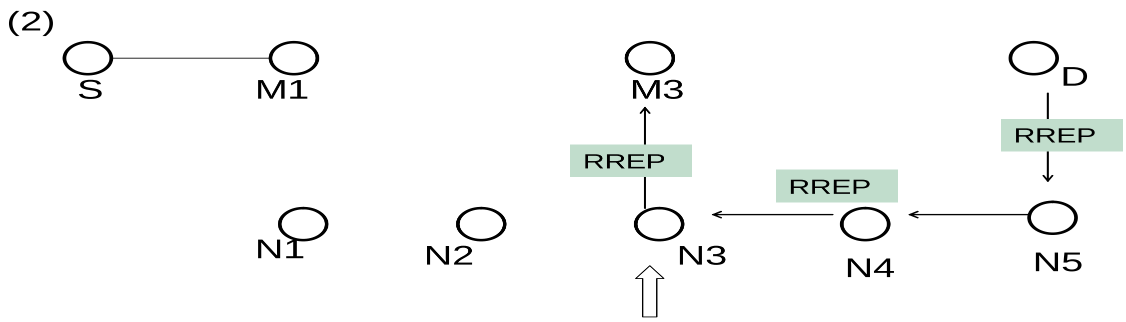
Node N3

• Receive RREQ(M3) ,then update routing table

Dst	Next Hop	Hop Count	SeqNo
S	M3	4	2

• Receive RREQ(M1) , discard it because it has neither larger SeqNo or smaller HopCount comapring to the routing table

(2)



Node N3

• Receive RREP ,then update routing table

Dst	Next Hop	Hop Count	SeqNo
S	M3	4	2
D	N4	3	2

• RREP is forward to node M3

Figure 9 Example of Reverse Direction Race Between a Local Repair Process and A Global Repair Process Repairing a Duplex Route in AODV

② Reverse Direction Race

The same direction race is a race that two repairs are trying to repair the same route with the same destination. Figure 9 shows an example of reverse direction race. In Figure 9, link M3-M4 breaks, and M3, M4 both detected the link break. The node M3 initiates a Local repair for the node D. The node M4 generates an RERR and sends it back to node D to initiate a global repair. Thus in the network two route repairs are processing to repair the same duplex route. Many redundant flooding of repair control message happen.

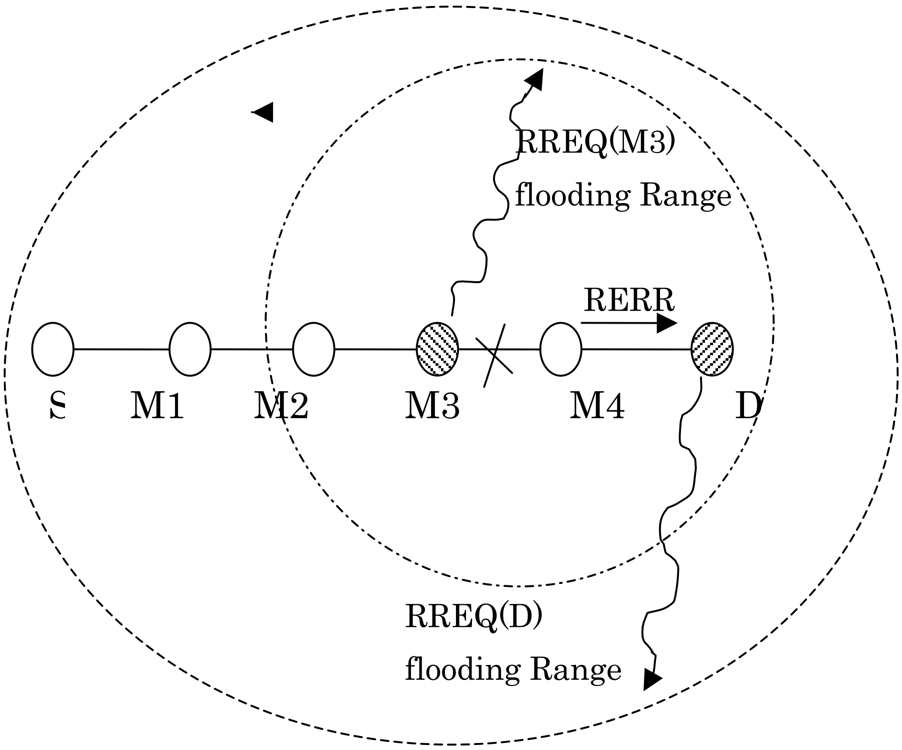


Figure 10 Image of Duplicate Overhead Generation When Reverse Direction Race Happened for Repairing a Duplex Route

CHAPTER 4 PROPOSAL OF LOCAL REPAIR OF DUPLEX ROUTE WITH RACE RESOLUTION

From Chapter 3, it is observed that a reverse direction race generates duplicate overhead of repair control messages, and a same direction race leads to various problems because the local repair whose broken link is nearest to the source node may fail. Thus, we propose a new Local Repair Scheme, where when a reverse direction race happens, the local repairs whose control message has reached a node earlier has higher priority. And when a same direction race happens, the local repair process whose broken link is nearest to the source node (i.e. farthest from the destination node) has higher priority than other local repairs. The detail principles of the proposal are given in Section 4.1 and their applications to AODV are described in Chapter 5.

4.1 Key Points

The policy of the proposed scheme is that

① For a reverse direction at a node, a local repair whose control message has reached the node later is given a low priority and is discarded.

② For a same direction at a node, only the local repair that can cover all the broken links is given a high priority and all the control messages of other repairs are given low priority and are discarded.

In the proposed scheme, each node has the following functions;

① History management of the repair processes in a Repair Control Table (RC Table)

② Race detection

③ Race resolution

The control message for a local repair has the following information:

① information to designate both the source node and the destination node
② information to designate the route freshness
③ information to designate the broken link's position in the route.

4.2 History Management of Repair Processes

When the node receives a repair control packet, the node takes out from the packet the information of the broken link's position in the route and stores it with the route identifier into the RC Table.

4.3 Race Detection

Referring to the RC Table, the node judges that there is a race if more than one repair processes are being executed for the same route. First, the node checks if a reverse direction race happened. For the reserve route it is repairing, if there is a RC entry which has a same ID and a same identifier of freshness with the repair control message, it judges that a reverse direction race happened. If a reverse direction happened, the node executes race resolution which is described in section 4.4. Next the node checks if there is a same direction race. Similar to reverse direction race detection, for the route which is being repaired if there is a RC entry which has a same ID and a same identifier of freshness, it judges that a same direction race happened and does same direction race resolution.

4.4 Race Resolution

In the reverse direction race resolution at a node, the priority of each repair is judged by its repair control message's arrival time. The local repair whose control message reaches the node later has a low priority and are discarded.

In the same direction race resolution at a node, the priority of each repair is judged by its broken link position in the route. If the broken link of a repair is farthest from the destination compared with all the repairs in the RC Table, then the local repair is given a higher priority. The routing table is updated based on this high priority repair and the repair control message is forward to other nodes. Otherwise, the repair control message is discarded without doing anything.

4.5 Control Message

The control message for a local repair has the following information:

① information to designate both the source node and the destination node

② information to designate the route freshness

③ information to designate the broken link's position in the route.

CHAPTER 5 IMPLEMENTATION INTO AODV

The challenges of local repair for duplex repair in AODV are shown in this chapter and then the format of RREQ, the RC table and the processing of the RREQ at nodes are described.

5.1 Challenges of Local Repair for Duplex Repair in AODV

① Unique RREQ recognition

For simplex route repair, <Originator IP Address, Originator Broadcast ID> is used to recognize a unique RREQ. But for duplex route repair, to do this we have to add a field into RREQ format for 'Originator IP Address'. For a lightweight format of RREQ, we propose an new unique RREQ recognition method which is described in 5.4.1.

② Loop problem

While implementing the proposed method into AODV, we have another problem to resolve as follows. In a local repair of duplex route for AODV, when the node which is upstream to the broken link in the same route receives this RREQ, it may update the reverse route to the source node. In this case, a loop will happen. To avoid this kind of loop problem, the RREQ is dropped in this case in our implementation as Figure 14 shown.

③ Update of whole route

To do this in our implementation, when local repair stats, the originator floods RREQ with source sequence number incremented by 1 and Destination sequence number incremented by 1. The image of the whole route update in case of local repair of duplex route for AODV.

④ Being friendly with global repairs

When race between global and local repairs happen, we have to cancel the racing local repair because global repair can cover all the broken links in the route. And because whenever a global repair failed, the source node will flood a RREQ again with only Source Sequence Number incremented. In this case although it may has a same Destination with other repairs which is racing with it, because it has a higher Source Sequence Number, the RREQ of the second try will not be ignored while updating a reverse route to the source. Our implementation is conscious of this in RREQ processing.

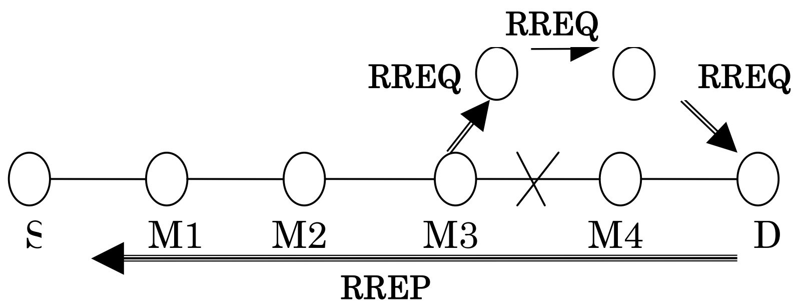


Figure 11 Image of Whole Route Update in Case of Local Repair of Duplex Route for AODV

5.2 Message Formats

5.2.1 Route Request (RREQ) Messages Format

The format of RREQ message is shown in Figure 12. The fields different from AODV are shaded.

OD hop count is a hop count from the broken link detector (i.e. originator) to the destination. It is assigned with the value of the hop count in the originator's routing table when a repair is initiated. In the case of the global repair, OD hop count is assigned to the value of INFINITY to be distinguished from a local repair.

5.2.2 Other Messages Formats

Other Messages (such as RREP, RRER) have the same formats with those in RFC 3561 [3].

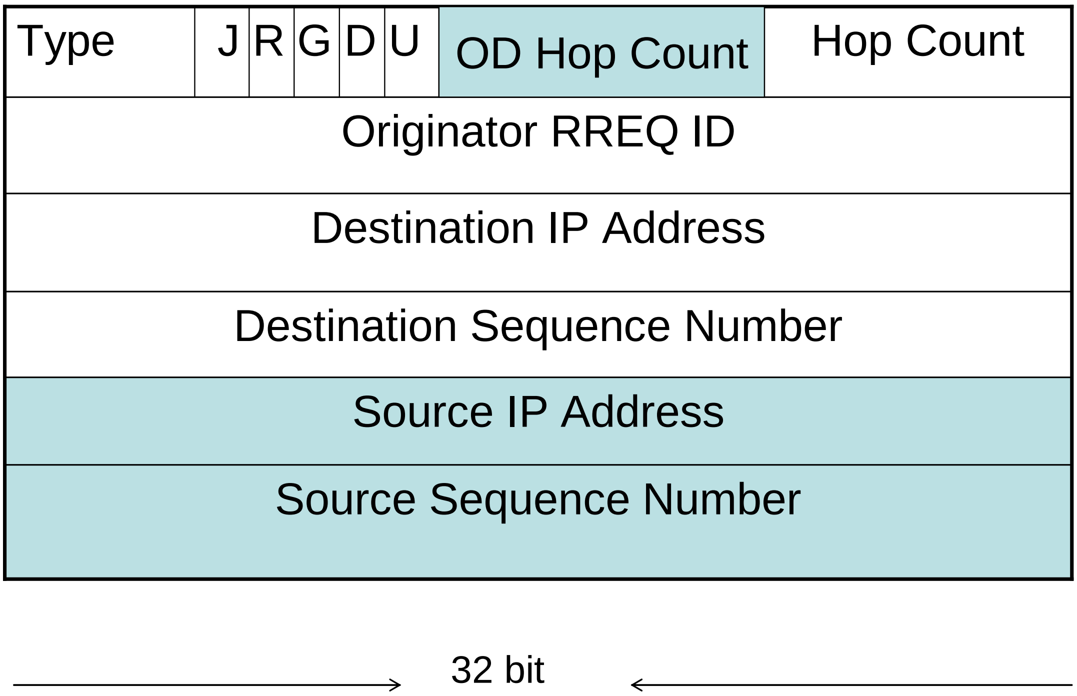


Figure 12 Format of RREQ in Proposed Method for AODV

5.3 Repair Control Table (RC Table)

In RC table, <destination IP address, destination sequence number> is used to determined a route and its freshness. OD hop count represents the broken links position in the route being repaired.

Destination IP Address	Destination Sequence No.	OD Hop Count

Figure 13 RC Table for AODV

5.4 Operation in Nodes

5.4.1 RREQ Processing

1. Unique RREQ

<Destination IP Address, Destination Sequence Number, OD Hop Count, Originator Broadcast ID> is used to recognize a unique RREQ. The node discards the RREQ if it has received before.

2. Upstream Drop

When a node receives a RREQ, if checks if it is in the same route and upstream of the link break. If it is, it must discard the RREQ as Figure 14 shows to avoid a loop generation.

If the RREQ has following field <S, value1(Src Seq No.), D, value2 (Dst Dseq No.)>, The nodes in the same route must have <S, value1-1> and <D, value2 -1> entries in the routing table. If the node has both of these entries in the routing table, it is considered to be in the same route with the RREQ originator.

Then if the node has a larger hop count to destination than OD hop count of the RREQ, the node is judged to be in the upstream of the broken link. It discard the RREQ.

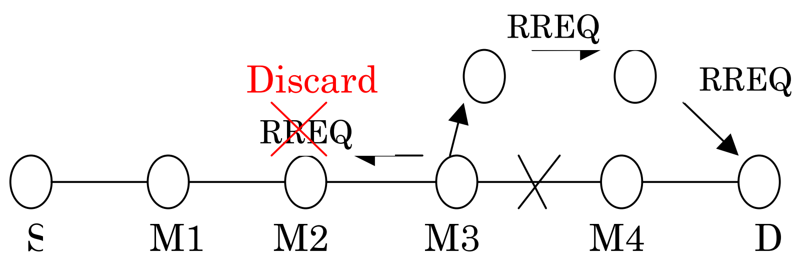


Figure 14 RREQ for Local Repair Discarded at Upstream Nodes

3. Race Resolution

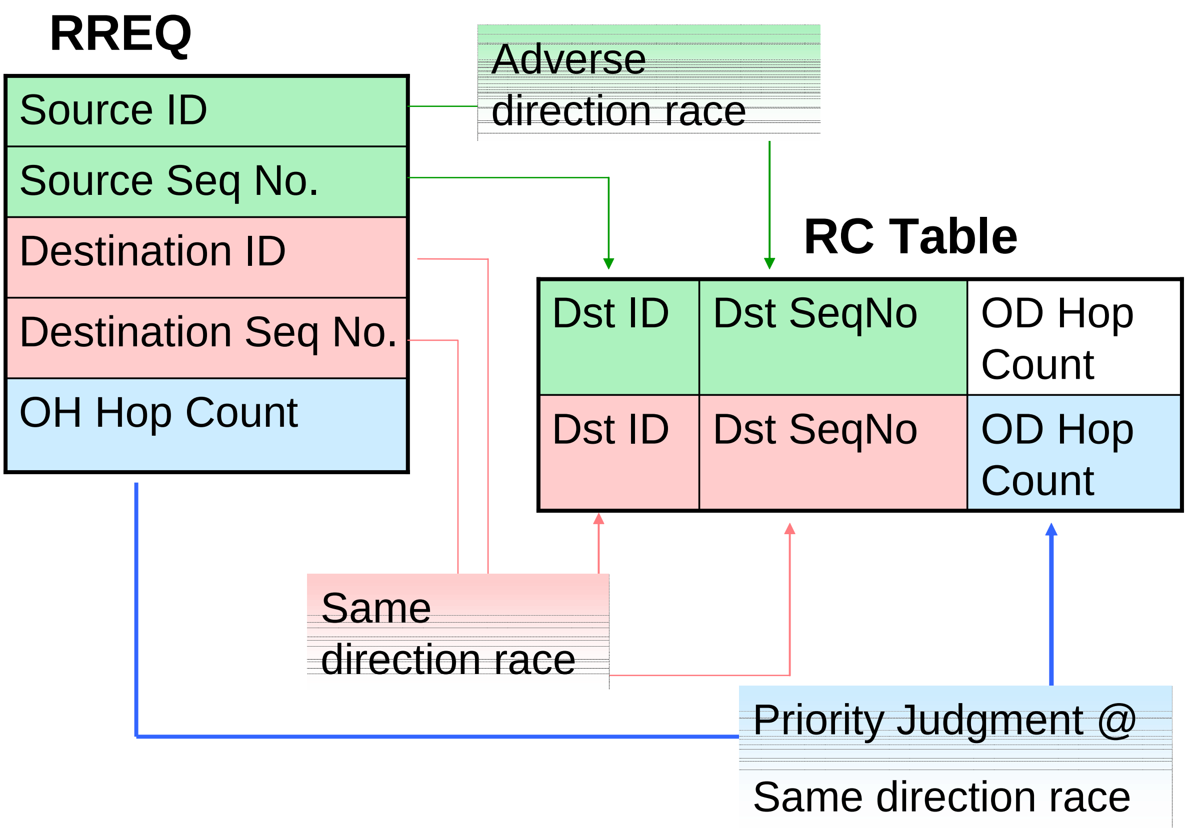


Figure 15　Race Judgment at Nodes

A node detects if race between repair processes happened and resolve the race when it happened. The image of the race resolution is shown in Figure 15. And its algorithm is described in Figure 16.

The node does reverse direction race detection and same direction race detection. If there is an RC table entry which has the same value which is the reverse route of the RREQ is repairing. Then reverse direction race happened and the RREQ is discarded in this case.

If the reverse direction races did not happen, then do the same direction race detection. If there is an RC table entry which has the same value which is the route of the RREQ is repairing. Then the same direction race happened. Then the priorities have to be assigned to those racing repairs. If the processing repair is a global repair, it has higher priority .And also if the process repair is a local repair but has a larger OD hop count than the racing repair which has been processed before, it has a higher priority. Otherwise, the repair has a lower priority. The RREQ is discarded in the case of it has lower priority. If the RREQ has a higher priority, then it can update corresponding routing table entry later.

After the judgment of races, routing table entry for the source node is updated. Besides those updating conditions in AODV, when the 'update_flag' has the value of 1(which is set at 20th line in Figure 16), the routing table entry is also updated.

It is also necessary to update the RC table. When the RREQ has a greater value of Destination Sequence Number comparing to that of the RC table entry, or if they has same Destination Sequence Number but the RREQ has a larger OD hop count, the RC is updated.

The last process where the node decides whether to forward the RREQ or to return an RREP has the same specification with RFC 3561 [3].

```
/*finding the corresponding RC table entry based on the destination IP
  address of the RREQ */
rc = rctable.rc_lookup(rq->rq_dst);

/* if no corresponding RC table entry exits,
   then a race did not happen. */
if (rc == 0) {
        rc = rctable.rc_add(rq->rq_dst, 1);
        rc->rc_dst_seqno = rq->rq_dst_seqno;
        rc->od_hop_count = rq->od_hop_count;
}
/* if a corresponding RC table entry exits */
else {
        rc0 = rctable.rc_lookup(rq->rq_src);

        /* reverse direction race judgment */
        if (rc0 != 0 && rc0->rc_seqno == rq->rq_src_seqno) {
                drop(p);
                return;
        }

    /* same direction race judgement */
        if (rc->rc_dst_seqno == rq->rq_dst_seqno) {
                /* the case that this RREQ is for a local repair
 or at the first try of a global repair */
                if ((rq->od_hop_count == INFINITY
                    && (rq->rq_src_seqno == rt0->rt_seqno)
                    | | (rq->od_hop_count != INFINITY) ){

 /* RREQ has a lower priority */
                        if (rq->od_hop_count < rc->od_hop_count){
                                drop(p);
                                return;
                }

 /* RREQ has a higher priority*/
                else if (rq->od_hop_count > rc->rc_hops ) {
                        rt_update_flag = 1;
                }
        }
}
```

Figure 16 Algorithm of RREQ processing with racing resolution

5.4.2 Other Operations

Processing of RRER, RREP and data packet are the same with AODV described in RFC 3561 [3].

CHAPTER 6 SIMULATION OF PROPOSED METHOD

6.1 NS2 simulator

NS2[17] is a discrete event simulator targeted at networking research. It provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. We use Ns simulator in our experimentation.

6.2 Conditions setting of mobile wireless network in NS2

- Creating scenarios

A scenario which define the nodes' movement in a simulation time is be described by an scenario file. The scenario file is created by a 'setdest' command. The options are listed as following.

-s speed type (uniform, normal)
-m minimum speed > 0
-M maximum speed
-P pause type (constant, uniform)
-p pause time (a median if uniform is chosen)
-n number of nodes
-x x dimension of space
-y y dimension of space

- Traffic

In ns, data is always being sent from one 'agent' to another. So the next step is to create an agent object that sends data from node n0, and another agent object that receives the data on node n1. #Create a UDP agent and attach it to node n0
Sending data

✧ creating a CBR traffic source:

```
set cbr [new Application/Traffic/CBR]
$cbr set packetSize_ 150
```

```
$cbr set interval_    0.1
```

- ✧ creating a UDP agent and attaching it into a node (the source node)

```
set udp [new Agent/UDP]
$ns attach-agent $node1 $udp
```

- ✧ attach the CBR traffic source to udp

```
$cbr attach-agent $udp
```

receiving data

- ✧ creating a sink agent and attaching it into the a node (the destination node)

```
set sink [new Agent/Null]
$ns_ attach-agent $node2 $sink
```

- ✧ connecting the sink agent and the udp agent

```
$ns_ connect $udp $sink
```

- Node Configure

A mobilenode consists of network components like Link Layer, Interface Queue, MAC layer, the wireless channel nodes transmit and receive signals from etc. An parameter setting example is given as following.

```
$ns_ node-config
    -addressingType flat or hierarchical or expanded
    -adhocRouting   AODV                            (ad-hoc routing protocol)
    -llType         LL                              (link layer type)
    -macType        Mac/802_11                      (MAC type)
    -propType       "Propagation/TwoRayGround"      (radio-propagation model)
    -ifqType        "Queue/DropTail/PriQueue"       (Interface queue type)
    -ifqLen         100                             (max packet in interface queue)
    -phyType        "Phy/WirelessPhy"               (network interface type)
    -antType        "Antenna/OmniAntenna"           (antenna type)
    -channelType    "Channel/WirelessChannel"       (channel type)
    -topoInstance   [new Topography]
    -agentTrace     ON or OFF
    -routerTrace    ON or OFF
    -macTrace       ON or OFF
    -movementTrace  ON or OFF
```

Setting of transmission range:

We specify the communication range of wireless nodes by set the receiving threshold in the network interface.

Phy/WirelessPhy set RXThresh_ <value>

The receiving threshold can be computed by indicting the transmission range using a C program named 'threshold' as following:

threshold -m <propagation-model> [other-options] distance

6.3 Implementation of Routing Protocols

In Ns simulator, there are some routing protocols implemented such as AODV, DSR. Since the implementation method which we proposed is for AODV, we modify the exiting AODV modules to our proposed method and compile it.

6.4 Simulation of Proposed Method

We set the parameters as following.

- Node configure Parameter Setting

Parameter	Value
Channel	Channel/WirelessChannel
propogation model	TwoRayGround
net interface	Phy/WirelessPhy
mac layer	IEEE 802.11
interface queue	Queue/DropTail/PriQueue
antenna	Antenna/OmniAntenna

Table 1 Node configure Parameters

- Queue Parameters Setting

Parameter	Value
Interface queue length	100

Table 2 Queue Parameters

- MAC Parameters Setting

Parameter	Value
CW Max	1023
Slot Time	20 [us]
SIFS	10 [us]
PLCP Data Rate	1 [Mbps]
RTS Threshold	240 [bytes]
Short Retry Limit	7
Long Retry Limit	4
Fragmentation Threshold	346 [bytes]

Table 3 MAC Parameters

- Routing Protocol Parameters setting:

Parameter	Valule
Routing Buffer Length	1000
Local Repair Wait Time	0.1[s]

Table 4 Routing Protocol Parameters

- Scenario Parameters

Parameter	value
Node moving area	400[m] x 400[m]
Node movement pattern	Random Way Point
Node number	110
Node transmission range	45[m]

Table 5 Scenario Parameters

- Traffic Parameters

Common parameters:

Parameter	Value
Transport Protocol	UDP
Numbers of Session	One source-destination pair
Traffic Source Type	Constant Bit Rate (CBR)
Packet Size	150 [bytes]

Table 6 Common parameters of Traffic

Traffic type 1: one-way traffic

Parameter	Value
CBR generation start time	0.5[s]
CBR generation end time	470[s]
Packets Generation Rate	12, 24, 40, 69 [kb/s]

Table 7 one-way traffic Parameters

Traffic type 2: two-way traffic

Parameter	Value
Source node CBR generation start time	0.5[s]
Source node CBR generation end time	470[s]
Destination node CBR generation start time	25[s]
Destination node CBR generation end time	470[s]
Source node Packets Generation Rate	12, 24, 40, 69 [kb/s]
Destination node Packets Generation Rate	12 [kb/s]

Table 8 two-way traffic Parameters

- Node speed:

Max speed	Min speed	Pause time
2[m/s]	1[m/s]	0[s]

Table 9 Mobility parameters

We did two experiments as following with above parameters

- Experiment 1 :
  - ✧ Node mobility:

    Two fix node S and D which has coordinate (230, 230) and (430, 430) respectively.
  - ✧ Traffic:

    One-way traffic from node S to node D (Traffic type 1)
  - ✧ Samples number: 9
- Experiment 2 :
  - ✧ Node mobility:

    Two fix node S and D which has coordinate (250, 250) and (410, 410) respectively.
  - ✧ Traffic:

    Two-way traffic between node S and node D (Traffic type 2)
  - ✧ Samples number: 1

CHAPTER 7 PERFORMANCE EVALUATION OF PROPOSED METHOD

7.1 Comparison Protocols

<u>Method Definition</u>:

Present repair methods:

- Global Repair (GR)

Proposed repair methods:

- Local Repair (LR)
  - ✧ Local Duplex route Repair (D-LR)
- Race Resolution (RR)
  - ✧ Same Direction Race Resolution (S-RR)
  - ✧ Reverse Direction Race Resolution (R-RR)

<u>Local Repair Start Condition</u>:

The originator node's hops to destination is no smaller than 2/3 of the whole route length.

<u>Comparison Protocols:</u>

Protocols of present research

- AODV with Only GR (GR AODV)

Protocols with unsound race resolution

- Proposed Method 1:

AODV with GR and D-LR (GR + D-LR AODV)

- Proposed Method 2

AODV with GR, D-LR and S-RR (GR + D-LR + S-RR AODV)

- Proposed Method 3

AODV with GR, D-LR, S-RR, R-RR (GR + D-LR + S-RR + R-RR AODV)

<u>Comparison Items</u>:

Experiment 1:

To evaluate same direction race resolution, the following protocols are compared.

- GR AODV
- Proposed Method 1
- Proposed Method 2

Experiment 2

To evaluate reverse direction race resolution, the following protocols are compared.

- GR AODV
- Proposed Method 1
- Proposed Method 2
- Proposed Method 3

7.2 Evaluation Metrics

<u>Race Occurrence Condition Evaluation Metrics:</u>

- Race Occurrence Count
  - ✧ Total Race Count (Race Count)
  - ✧ Race among Global Repair and Local Repair Count (GRLR Count)
  - ✧ Race among Local Repairs Count (LRLR Count)
  - ✧ Race among Global Repair and more than one Local Repair Count (GR-multiLR Count)

<u>Race Resolution Evaluation Metrics</u>:

- RREQ Overhead During Race
  - ✧ Local Repair Overhead per Race
  - ✧ Average Overhead per Local Repair During Race

- Route Repair Failure(Cancellation) States During Race
  - ✧ GR failure Count During GRLR Race
  - ✧ LR Cancellation Count During GRLR Race
  - ✧ High Priority LR Failure Count During LRLR Race
  - ✧ Low Priority LR Cancellation Count During LRLR Race
  - ✧ LR Cancellation Ratio During GRLR Race
    LR Cancellation Count / GRLR Race Count
  - ✧ Low Priority LR Cancellation Ratio During LRLR Race
    Low Priority LR Cancellation Count / LRLR Race Count

Network Performance Evaluation Metrics:

- Repair Overhead
  (RREQ + RRER) x hops
- Data Packet Delivery Rate
  Destination arrival number / Source generation number
- Average Data Packet Transmission Delay
  Destination arrival time – source generation time

7.3 Simulation Results

- Results of experiment 1

  Race occurrence Count :
  Figure 17-20

  Race resolution:
  Figure 21-30

  Network performance:
  Figure 31-33

- Results of experiment 2

  Race occurrence Count :
  Figure 34-35

  Race resolution:

Figure 36-37

Network performance:

Overhead: Figure 38-39

Data Packet Delivery: Figure 40-41

Data Packet Delay: Figure 42-43

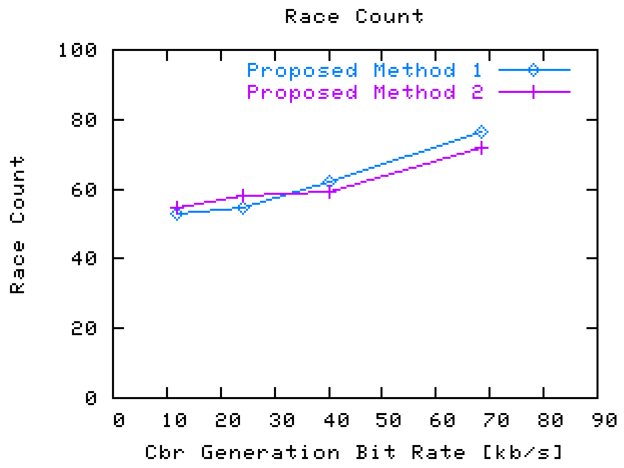


Figure 17 Result of Experiment1: Race count

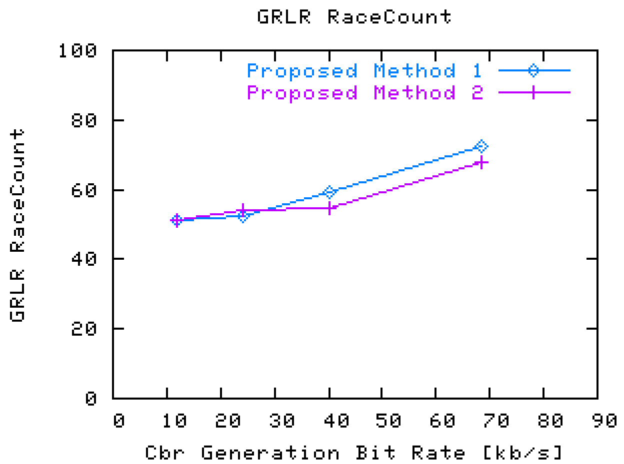


Figure 18 Result of Experiment1: GRLR Race count

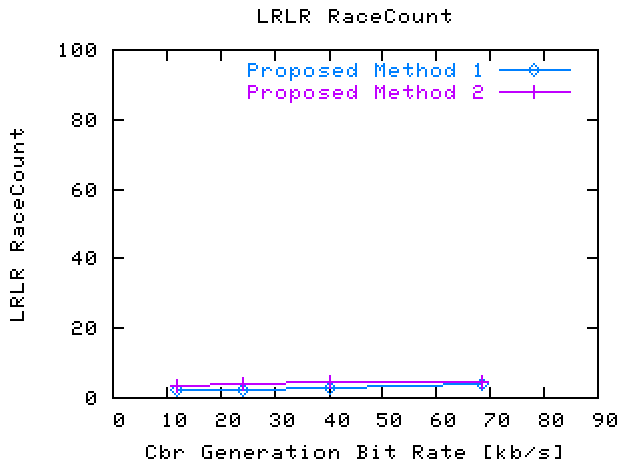


Figure 19 Result of Experiment1: LRLR Race count

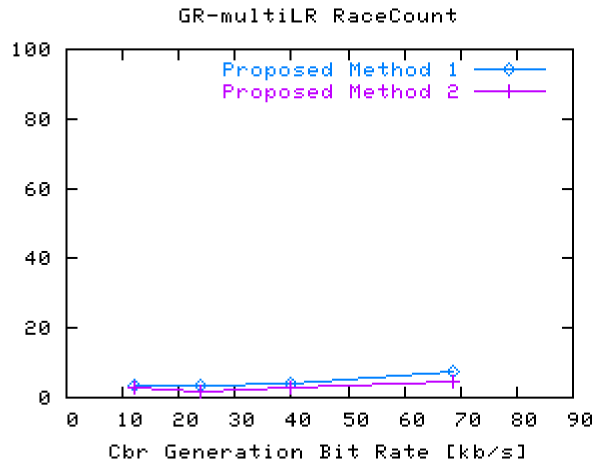


Figure 20 Result of Experiment1: GR-multiLR Race count

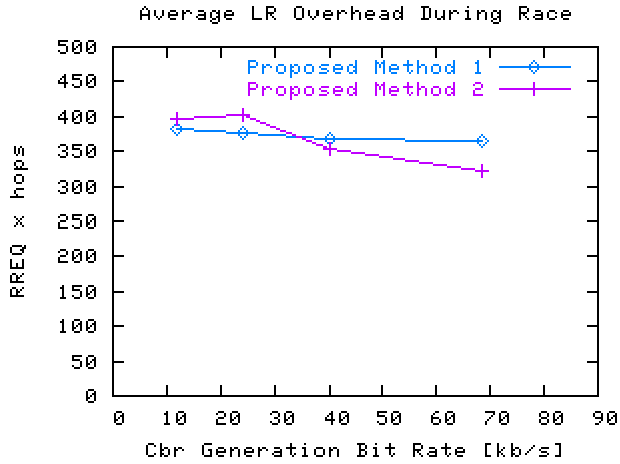


Figure 21　Result of Experiment1: Local Repair Overhead per Race

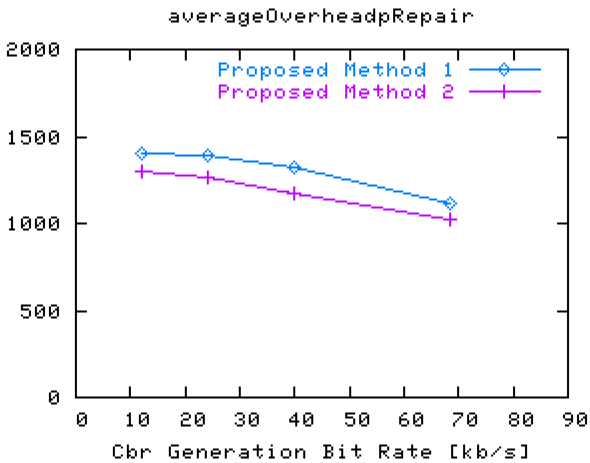


Figure 22　Result of Experiment1: Average Overhead During Race

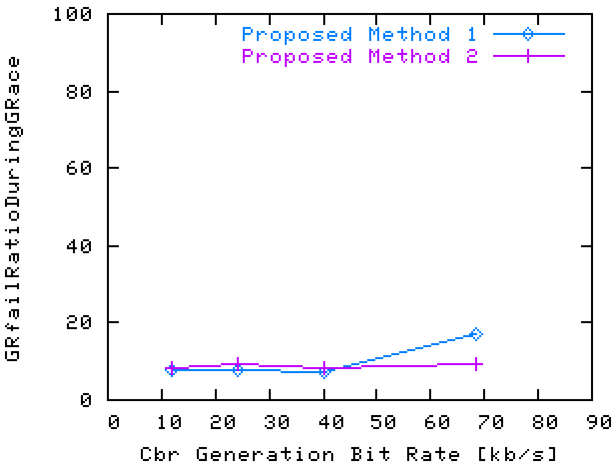


Figure 23　Result of Experiment1: GR failure Count During GRLR Race

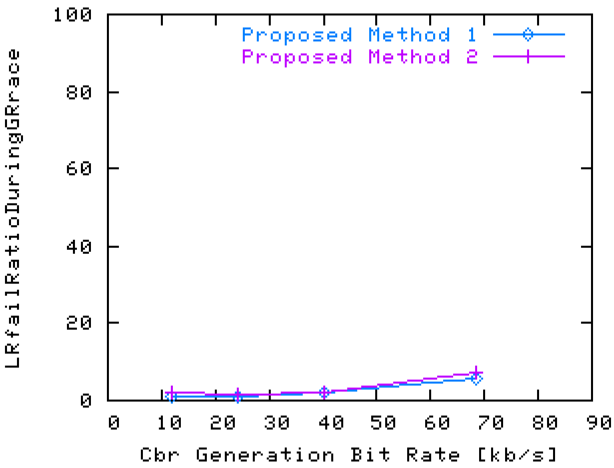


Figure 24　Result of Experiment1: LR Cancellation Count During GRLR Race

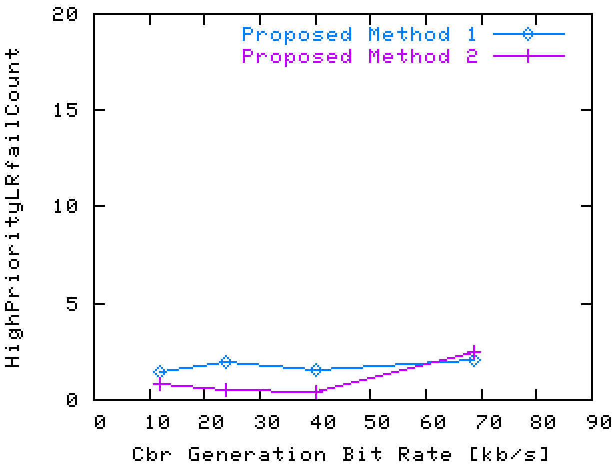


Figure 25　Result of Experiment1: High Priority LR failure Count During LRLR Race

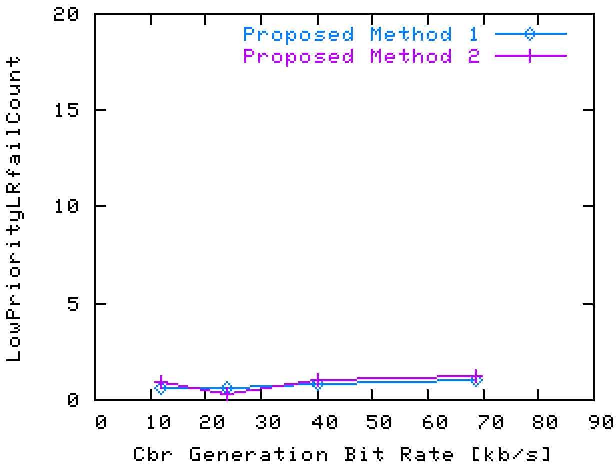


Figure 26　Result of Experiment1: Low Priority LR Cancellation Count During LRLR Race

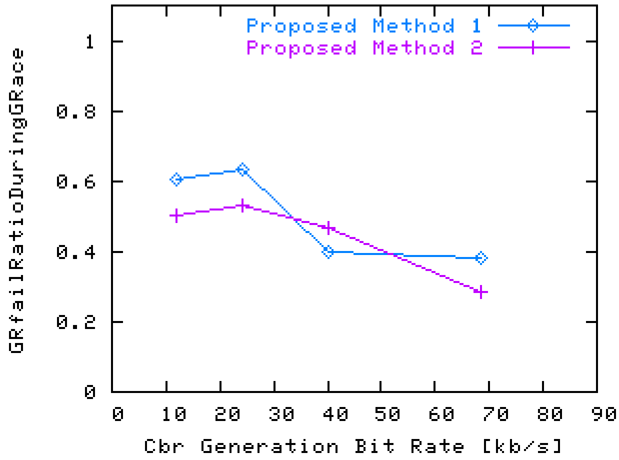


Figure 27 Result of Experiment1: GR failure Ratio During GRLR Race

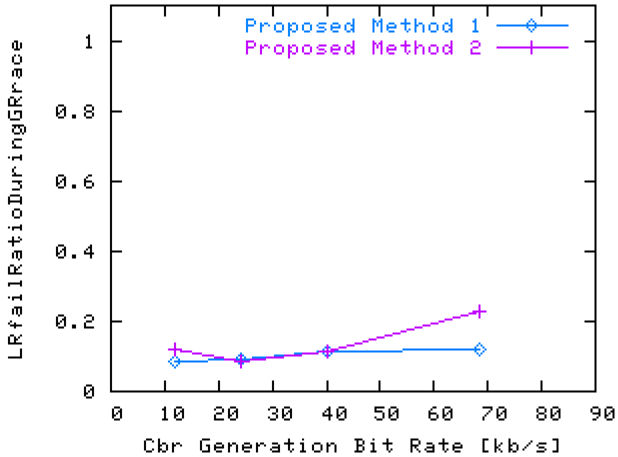


Figure 28 Result of Experiment1: LR Cancellation Ratio During GRLR Race

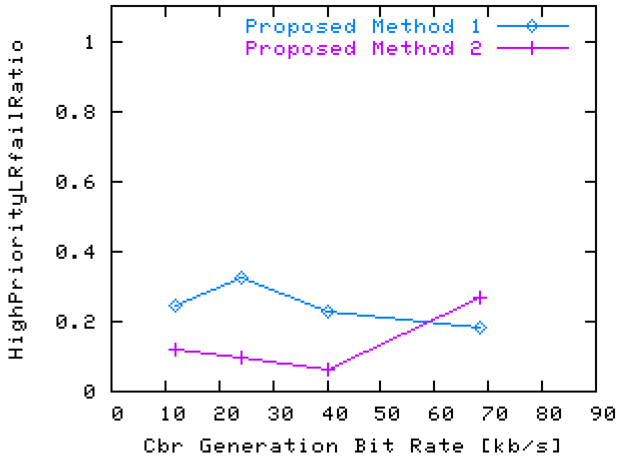


Figure 29 Result of Experiment1: High Priority LR Cancellation Ratio During LRLR Race

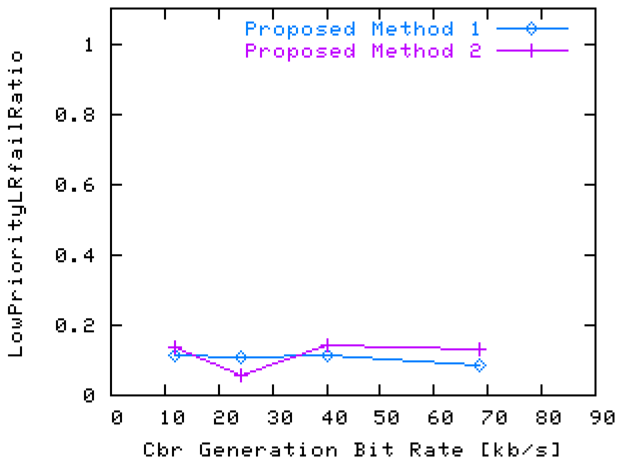


Figure 30 Result of Experiment1: Low Priority LR Cancellation Ratio During LRLR Race

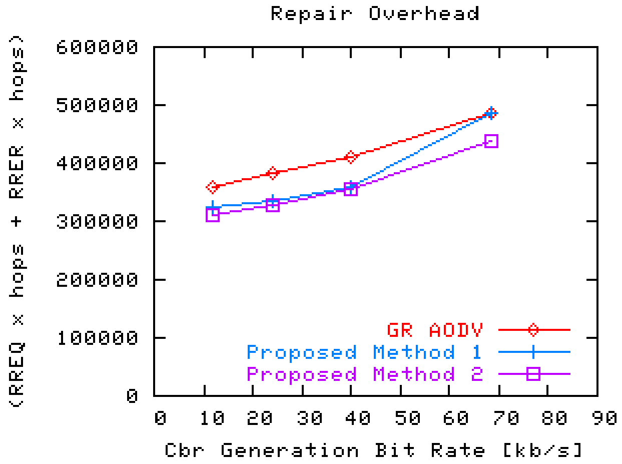


Figure 31 Result of Experiment1: Total Repair Overhead

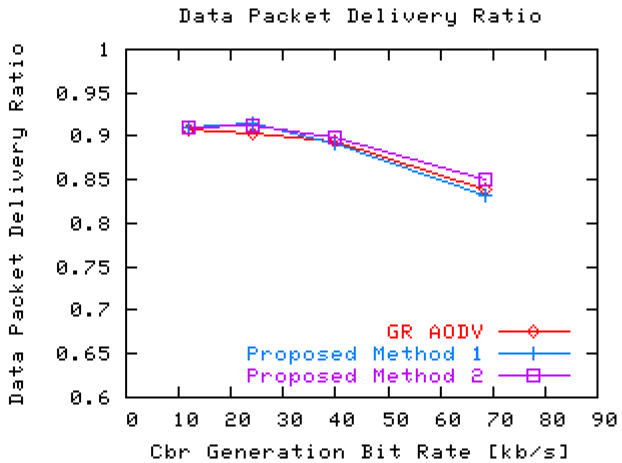


Figure 32 Result of Experiment1: Data Packet Delivery Ratio

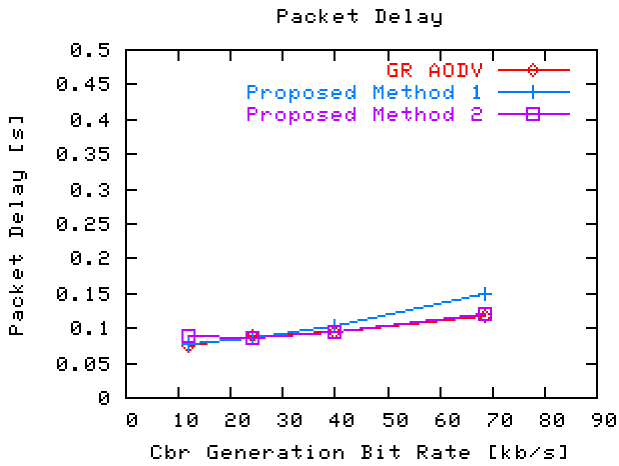


Figure 33 Result of Experiment1: Data Packet Delay

- Results of experiment 2

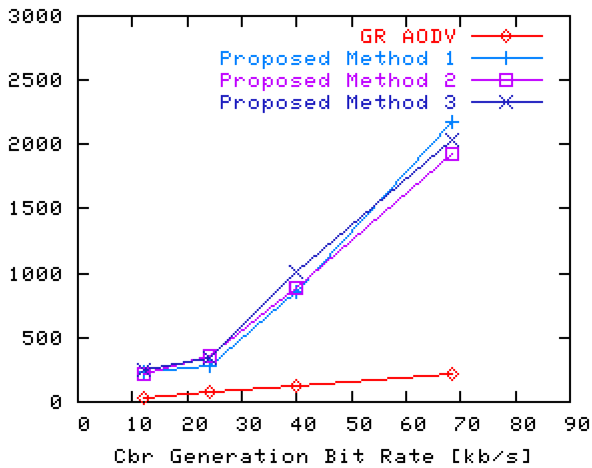


Figure 34 Result of Experiment 2: Total Reverse Race Count

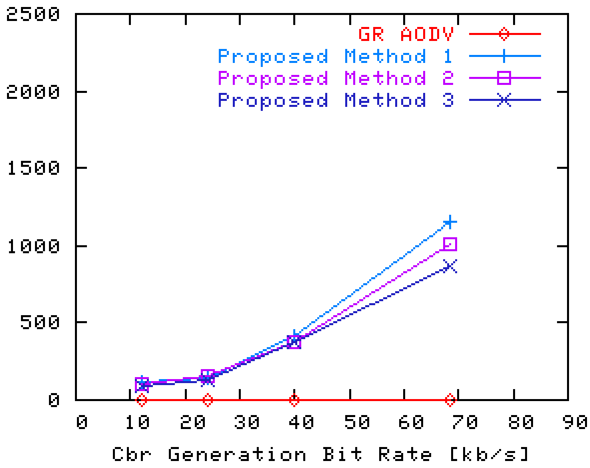


Figure 35 Result of Experiment 2: LRLR Reverse Direction Race Count

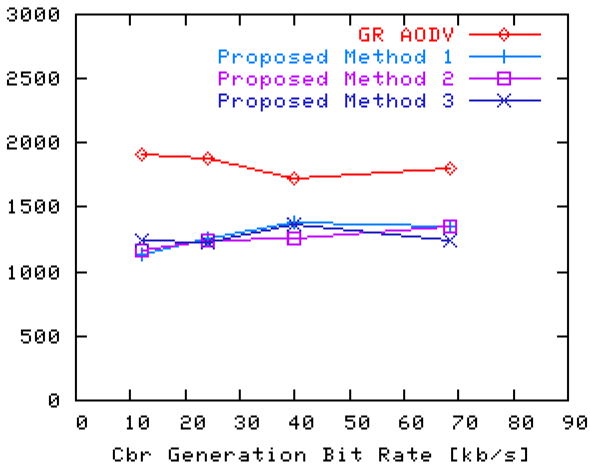


Figure 36 Result of Experiment 2: Average Overhead per Reverse Direction Race

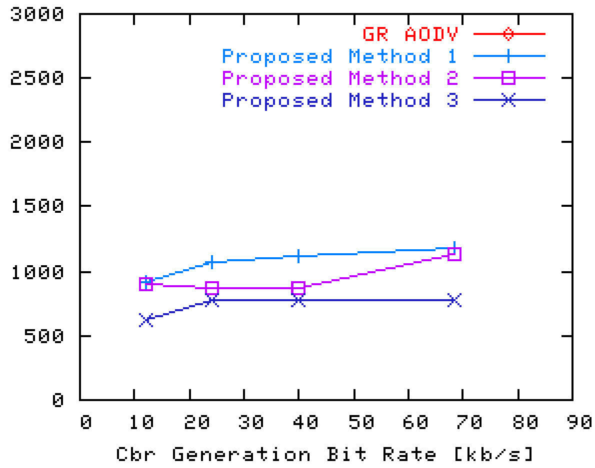


Figure 37　Result of Experiment 2: LRLR Reverse Race

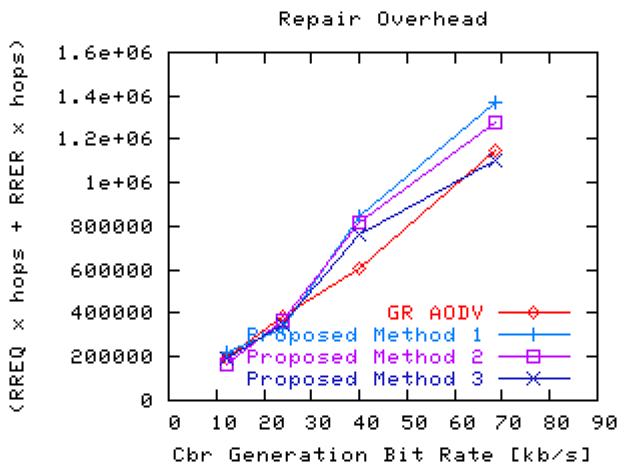


Figure 38　Result of Experiment 2: S→D Route Repair Overhead

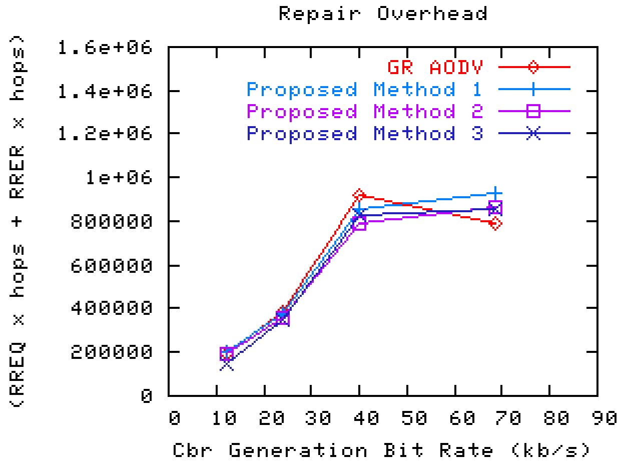


Figure 39 Result of Experiment 2: D→S Route Repair Overhead

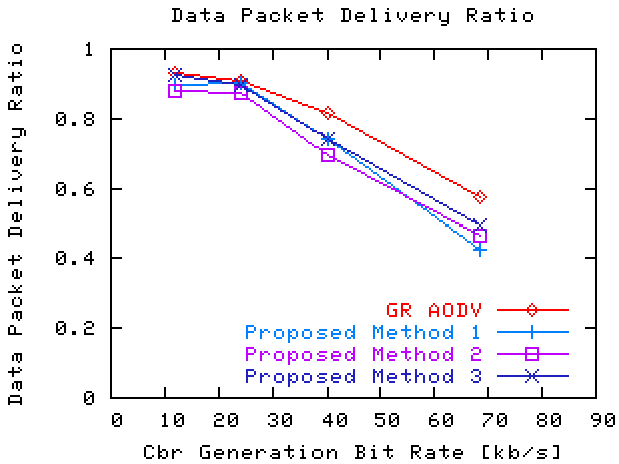


Figure 40 Result of Experiment 2: S→D Data Packet Delivery

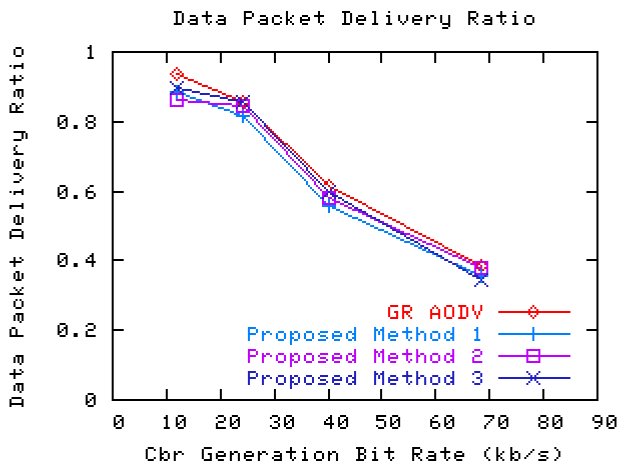


Figure 41 Result of Experiment 2: D→S Data Packet Delivery

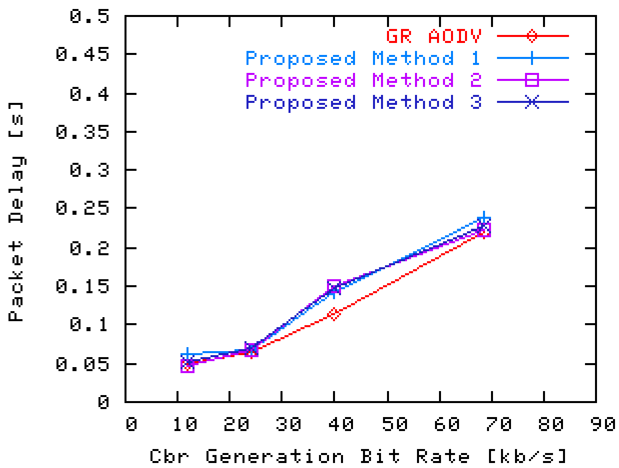


Figure 42 Result of Experiment 2: S→D Data Packet Delay

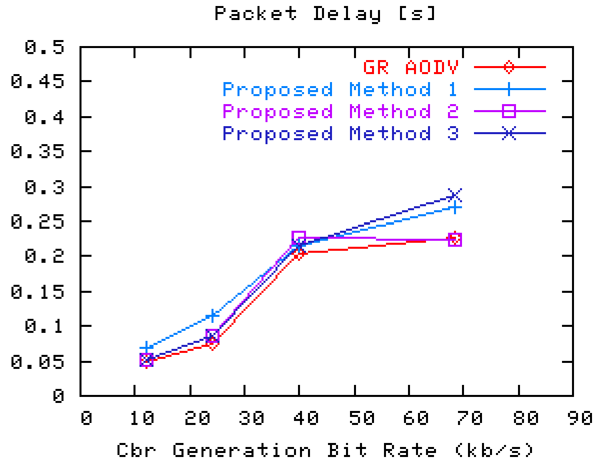


Figure 43 Result of Experiment 2: D→S Data Packet Delay

7.4 Comparison and Evaluation

<u>Comparison in experiment 1</u>:

From race occurrence count results of experiment 1 (Figure 17-20), we can see in proposed method 1, same direction races happen more frequently.

Figure 21-22 shows that proposed method 2 has the smallest repair overhead while racing.

Figure 23-30 shows that when races happen, proposed method 2 has higher possibility to cancel the redundant local repair and has higher success possibility of the whole route repair in comparison to the proposed method 1 which has no race resolution method.

Figure 31-33 shows the network performance result. The proposed method 2 has the lowest total overhead for repairs and has higher data delivery ratio than other comparison protocols. All the proposed protocols have higher data delay because they created longer route.

<u>Comparison in experiment 2:</u>

Figure 34-35 shows that all proposed method cause reverse direction race more frequently and most of the races are between local repairs.

Figure 36-37 shows that all proposed methods have lower average total overhead while reverse direction race happens. Proposed method 3 which has a reverse direction race resolution has the lowest overhead while a reverse direction race between local repairs happens.

Figure 38-39 shows that proposed methods has lower overhead for route repairs in the cases of source CBR generation rates are low. In the cases of CBR generation rates are high, GR AODV has the lowest overhead, and proposed method 3 is the best among proposed methods.

Figure 40-41 shows all proposed methods has lower data delivery ratio than GR AODV. The proposed method 3 has the best result among proposed methods.

Figure 42-43 shows all proposed methods have larger data transmission delay results in comparison with GR AODV

Evaluation:

In the one-way traffic evaluation experiments, proposed methods was shown to have the smallest overhead of repair control packets for local repairs and has the highest probability of route repair success. The proposed method with same direction resolution has the highest network performance. In the two-way traffic evaluation experiments, it is shown that the proposed method which has a reverse direction race resolution has the smallest overhead of repairing control packets compared to other proposed local repair methods. In all these experiment results, the proposed methods restrict the redundant flooding of repair control messages and avoid the local repair failure caused by the races.

CHAPTER 8 CONCLUSION AND FUTURE WORK

8.1 Conclusion

In this paper, race problems with MANET are analyzed. Race may happen in the case where the route repair control function is decentralized and multiple route repairs are processed independently. Races result in redundant overhead of repair control packets. Races also have bad effect on the whole route repairs especially in the case of duplex route repairing.

The local repair is a method to reconstruct a route with smaller overhead in comparison with the global repair in repairing a long route. On the other hand, the global repair can achieve more optimal route with smaller number of hops. AODV adopts the combination of local and global repairs for repairing simplex routes with small overhead but AODV does not have a function to resolve the race problems and it can repair only simplex routes with its local repair method, which leads to low network performance in the case of a race for duplex route failures. In order to overcome these defects, we proposed a new local repair scheme with race resolution, which can correctly repair duplex route rapidly with small overhead of control packets in the network. Considering the present local repair method of AODV is not suitable for repairing duplex routes, we also proposed an implementation method of the proposed scheme into AODV.

We have conducted two types of evaluation experiments to clarify the effectiveness of the proposal: one-way traffic experiment for the evaluation of simplex route repair and two-way traffic experiments for the evaluation of duplex route repair. In the one-way traffic evaluation experiments, proposed method with race resolution was shown to have the smallest overhead of repair control packets for local repairs and have the highest probability of route repair success. In the two-way traffic evaluation experiments, it is shown to have the smallest overhead of repairing control packets compared with other local repair methods. proposed methods with race resolutions restrict the redundant flooding of repair control message and avoid the local repair failure caused by the races.

8.2 Future Work

The proposed resolution method for race problems with duplex routes could be enhanced to improve its efficiency in terms of reducing the control packet overhead and also the repairing time to minimize the suspension of user packet transmission. Furthermore, the implementation of the proposed method into typical routing protocols for MANET other than AODV is considered another future work.

References

[1] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, “Scenario-based performance analysis of routing protocols for mobile ad-hoc networks”, Proc.ACM/IEEE MOBICOM’99, pp.195-206, Aug. 1999.

[2] D.B. Johnson, D.A. Maltz, “Dynamic Source Routing in Ad Hoc Wireless Networks”, in Mobile Computing, ed. By T.imielinski and H.korth, Kluwer Academic Publishers, 1996.

[3] RFC 3561: “Ad hoc On-Demand Distance Vector (AODV) Routing”

[4] Hridesh Rajan, "Effects of Applying Mobility Localization on Source Routing Algorithms for Mobile Ad Hoc Network", In the proceedings of IEEE Symposium on Computers and Communication (ISCC) 2003, July 2003.

[5] C.-K.Toh, “Associativity-based routing for ad-hoc mobile networks”, Wireless Pers.Commun.J., vol.4, no.2, pp.103-139, March 1997.

[6] R. Dube, C.D.Rais, K.-Y. Wang, and S.K. Tripathi, “Signal stability-based adaptive routing(SSA) for ad hoc mobile networks,” IEEE Personal Communi., pp.36-45, Feb.1997.

[7] W. Su and M. Gerla, “IPv6 flow handoff in ad-hoc wireless networks using mobility prediction,” Proc. IEEE GLOBECOM, pp.271-275, Dec.1999.

[8] Srinath Perur, Abhilash P. and Sridhar Iyer, “Router handoff: a preemptive route repair strategy for AODV”, IEEE Intel. Conference on Personal Wireless Computing, New Delhi, Dec, 2002.

[9] Morteza Maleki, Karthik Dantu and Massoud Pedram, “Power-aware Source Routing Protocol in Mobile Ad Hoc Networks”, Proceedings of International Symposium on Low Power Electronics and Design (ISLPED) 2002.

[10] T. Goff, N. B. Abu-Ghazaleh, D. S. Phatak and R. Kahvecioglu, “Preemptive Maintenance Routing in Ad Hoc Networks”, MOBICOM 2001, Rome, Italy, July 2001

[11] C.K.Toh. Translated by Kozo Keikaku engineering INC “Ad Hoc Mobile Wireless Networks(in Japanese)” Kozo Keikaku engineering INC, 2003

[12] Ki-Hyuang KIM and Hyun-Gon SEO, “The Effects of Local Repair Schemes in

AODV-Based Ad Hoc Networks", IEICE TRANS.COMMUN.vol.E87-B,No.9 Sep 2004

[13] Yusuke Sakurai, Yasuyuki Nishibayashi and Jiro Katto,"A Proposal to Improve TCP Performance over Ad Hoc Network using AODV"(in Japanese), IEICE General Conference 2003, March 2003

[14] Hideaki Matsue, Masahiro Morikara, Akio Sato and Kazuji Watanabe, "High-speed Wireless Access Technologies", IEICE, 2004

[15] Mattew S. Gast , Translated by Tadanori Mizuno, Takasi Watanabe, Susumu Isihara and Hiroshi Mineno,"802.11 Wireless Network Management", Ohmsha, 2003

[16] W. Navidi and T. Camp, "Stationary Distributions for the Random Waypoint Mobility Model", IEEE Transactions on Mobile Computing, vol. 3, no. 1, pp. 99-108, January-March 2004.

[17] The Network Simulator ns-2 Documentation: http://www.isi.edu/nsnam/ns/ns-documentation.html

Publications List

1. Meng Zhang, Fumitaka Nakamura, Yasushi Wakahara, "Local Route Repair for Ad Hoc Network", IEICE Society Conference 2004, September 2004
2. Meng Zhang, Fumitaka Nakamura, Yasushi Wakahara, "Efficient Local Repair with Race Resolution for Duplex Route in Ad Hoc Networks", IEICE 1st Ad Hoc Network Workshop, January 2005
3. Meng Zhang, Fumitaka Nakamura, Yasushi Wakahara, "Efficient Local Repair with Race Resolution for Duplex Route in Ad Hoc Networks", IEICE General Conference 2005, March 2005