

# Quick Local Repair Scheme using Adaptive Promiscuous Mode in Mobile Ad Hoc Networks

Joo-Sang Youn<sup>1</sup>

<sup>1</sup>Dep. of Electronics Engineering, Korea University, SEOUL, KOREA  
Email: ssrman@widecomm.korea.ac.kr

Ji-Hoon Lee<sup>2</sup>, Doo-Hyun Sung<sup>3</sup> and Chul-Hee Kang<sup>1</sup>

<sup>2</sup>Communication Lab. Samsung Advanced Institute of Technology, YONGIN, KOREA  
<sup>3</sup>Mobile Communication Technology Research Lab. LG Electronics, ANYANG, KOREA  
Email: vincent.lee@samsung.com, {ssungdoo, chkang}@widecomm.korea.ac.kr

**Abstract**—In mobile ad hoc networks (MANETs), there is frequently disconnected a route consisting of multi-hop from a source to a destination because of the dynamic nature such as the topology change caused by nodes' mobility. To overcome this situation, existing routing protocols for MANETs have performed route repair scheme to repair the disconnected route. However, existing reactive routing protocols have the problem which is that a source node unnecessarily performs re-discovers the whole path when just one node moves, even if the rest of path needs not to be re-arranged. Therefore, the time for re-discovery of the whole path may often take too long. To solve the problem, we propose a new local repair scheme using promiscuous mode. Our scheme is mainly composed of two parts: adaptive promiscuous mode and quick local repair scheme. Adaptive promiscuous mode is to repeat the switching processes between promiscuous mode and non-promiscuous mode to overcome energy limit caused by using promiscuous mode in overall time and quick local repair scheme is to fast perform the local re-route discovery process with the information of the active connection in the local area acquired by promiscuous mode. With simulation in the various number of connection, We demonstrate the better network performances achieved with the proposed schemes as compared with AODV as reference model that do not provide local repair scheme.

**Index Terms**—Local repair schemes, mobile ad hoc networks (MANETs), promiscuous mode.

## I. INTRODUCTION

Mobile Ad hoc Networks (MANETs) is a self-configuring network of mobile nodes, which also function as routers, connected by wireless links. Nodes are free to move randomly and organize themselves into local area network arbitrarily [1]. Thus, the topology of networks may change rapidly and unpredictably. Under these circumstances, it is significant how efficiently routing protocols manage the fresh and valid route

information.

There are two approaches in terms of how routing protocols manage and provide routing information: proactive and reactive [2]. Proactive routing protocols, called table-driven routing protocols, maintain consistent routing table in each node. DSDV (Destination-Sequence Distance-Vector Routing) [4] and OLSR (Optimized Link State Routing) [8] are well-known proactive routing protocols. This approach, however, has a drawback that many route updating messages flood into whole network to maintain up-to-date network information. Then, in spite of advantage to low latency getting a route, proactive approach is not proper for ad hoc environments with mobile nodes. On the other hand, reactive routing protocols, called on-demand routing protocols, do not maintain any of consistent routing information. Instead, they instantly create routes to a destination node only when a source node wants to communicate with a destination node. Examples of reactive routing are DSR (Dynamic Source Routing) [3, 6] and AODV (Ad hoc On-demand Distance Vector) [5]. Compared to proactive approaches, these protocols eliminate the need to periodically flood the network with table update messages which are required in a table-driven approach. Hence, reactive routing protocols are considered as a proper routing approach for MANET even though the instant route construction induces more delay than consistent routing information in proactive routing protocols.

Even if reactive routing protocols are suitable for unpredictable topology changes in MANET, they have weak points. Because the route maintenance mechanism of the protocols does not locally repair a broken link, they have to reconstruct the whole route in most cases, even though the route error has been caused by just one node's problem. Route reconstruction from a source node to a destination node adversely affects the network performance. Not only long latency for repairing the broken route but also excessive flooding of route request messages are induced by inefficient route repair mechanisms of existing routing protocols. Therefore,

---

Based on "A Local Repair Scheme with Adaptive Promiscuous Mode in Mobile Ad Hoc Networks", by Doo-Hyun Sung, Joo-Sang Youn, Ji-Hoon Lee and Chul-Hee Kang, which appeared in the Proceedings of 1st Intl. Conf. MSN 2005, Wuhan, China, Dec. 13- 15.

even though the protocols perform well in static and low-mobility ad hoc environments, the performance degrades rapidly with increasing mobility.

In ad hoc environment with the high mobility of nodes to solve the inefficiency of route repair and improve network performance, we propose a quick local repair scheme using adaptive promiscuous mode in a reactive (on-demand) approach. In the proposed scheme, if an established route is broken, it is locally recovered by aid of adjacent nodes which operate in promiscuous mode around the breakage area. And nodes in the networks can switch the receiving mode, i.e. promiscuous mode and non promiscuous mode, based on the pre-defined criteria. The proposed scheme has two advantages compared with the existing local repair schemes. First, adaptive promiscuous mode in which nodes switch operation between promiscuous and non-promiscuous mode reduces the overheads of promiscuous mode compared to general promiscuous mode. Second, the time of route repair about route error is considerably reduced by quick local repair mechanism.

The rest of this paper is organized as follows. Section II reviews existing schemes. We introduce the motivation of our work in section III. In section IV, the proposed scheme is described. We evaluate the proposed scheme based on simulation results in section V. Finally, we draw out conclusions in section VI.

## II. RELATED WORKS

We introduce existing local route repair schemes. Reactive routing protocols are advantageous to cope with random movement of nodes. Most of them, however, do not have efficient route repair algorithms. When a node detects the route error, it sends a route error message back to the source node, which leads the source node to activate route discovery process. To solve the inefficiency problem of route repair in reactive routing protocols, there were several studies to locally recover the disconnected route.

LRR scheme [9] assumes that the link error is caused by the relative movement of only one node on the route. A “neighbor node” is defined as a node which is on the route from the source to the destination and is in the immediate vicinity of the moved node. In other words, a neighbor node is a former one of moved node on the route. The aim of this scheme is to patch the route between the two nodes of the broken path through some other link or node. The zone in which the route-repair packet propagates is defined as the “request zone.” In LRR, the TTL field of the IP packets is used to limit the request zone to two hops. The neighboring node in the direction of the source initiates the patch up since it is first to recognize that the route is broken. The neighbor node broadcasts a route repair packet with TTL=2 so as to reach a latter node of moved node and then recover the route error locally.

AODV-BR [10] is a modified protocol from AODV literally. The basic route discovery process has not been changed. A source node sends a RREQ packet to a destination node, and the destination reply to RREQ as a

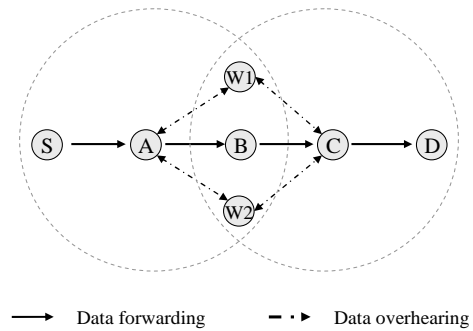


Figure 1. Example of WAR

RREP packet. The different thing is that every node in the network operates in promiscuous mode. When a RREP packet comes back to the source node, a neighboring node which is not part of the route overhears a RREP packet. And then it records the node which transmitted a RREP packet as the next hop to the destination in its alternate route table. Likewise this, all nodes existing beside of the route updates their alternate route table after overhearing RREP packet. After these operations, the members of the route and neighboring nodes organize a mesh structure. The original route from the source to the destination is called as primary route, and the rest of the routes are called alternate route. Within these mesh structures, a data packet is delivered via an alternate route when the primary route is disconnected.

Witness-Aided Routing (WAR) [7] has a similar concept compared to AODV-BR. Like AODV-BR, WAR allows nodes to operate in promiscuous mode. A witness is defined as a host which can overhear a transmission that is not destined to it. Fig.1 shows how witnesses participate in the routing process. Both node W1 and W2 hear node A’s transmission to node B, which makes them potential active witnesses of node A with respect to the packets sent from node A to node B. At this point, they will wait to see if node B attempts to deliver the packet to node C, which would mean that node B received it from node A. If that is the case, their role with respect to the packets sent from node A to node B reduces to sending an acknowledgement to node A (to avoid an error in case node A could not hear node B’s transmission to node C). If neither W1 nor W2 hear node B’s transmission to node C, they conclude that the packets from node A to node B failed to reach node B. In this case, they will both attempt to deliver the packet directly to node C, although, indirectly, they target node B as well. Since node W1 and node W2 do not necessarily have a way to communicate with each other and avoid contention, they will ask node C for arbitration before sending the packet. If node C rejects their request, it means that it has already received the packets from node B and their role reduces to sending the acknowledgement to node A.

## III. MOTIVATION

The motivation has been derived from the absence of efficient local route repair scheme in reactive routing protocols. There were several studies to resolve the

problem. Following section describes the problem of existing route repair, i.e. route re-construction, in detail. Then we show the limitations of existing local route repair schemes. Finally, the basic concepts of proposed local route repair scheme will be briefly introduced.

*A. Problems of Route Reconstruction*

As MANETs consist of nodes under highly dynamic environment, where frequent route changes are expected, routing protocols play a key role on network performance. They have to reflect the dynamic topology and provide the best route to the destination. Therefore the reactive route construction is indispensable in this circumstance. However, it is also critical issue to recover the route error caused by nodes' mobility. To resolve the problem, it is the best way that a source node didn't detect the route break.

Unfortunately, reactive routing protocols which are mostly used, e.g. DSR or AODV, do not have any efficient local repair algorithms. In DSR, nodes can record multiple route information to the same destination in route caches. When the route breakage happens, the packet transmissions may not be stopped by aid of multiple route information of intermediate nodes. However, when route break happens and there is no additional route information at the node that detects the route break, the route error message is sent to the source node. Then the source node tries to re-discover a new route by activating route discovery process. This can not be considered as local route repair. In AODV, there exists a local repair process. This process let the node detecting route error find a new route from itself to destination. Its drawback is that the route can be longer than previous one. In addition, the local repair process is activated limitedly when the route error happens not far from destination node. Thus, the local repair can not be generally used. As a result, both DSR and AODV rely on route reconstruction from source to destination in most cases.

*B. The Adverse Effects of Route Reconstruction*

The adverse effect of route reconstruction from source to destination is divided into two points [9, 11]. First, the time is wasted too much. Consider the example in Fig. 2, after the route creation from node S to node D, the

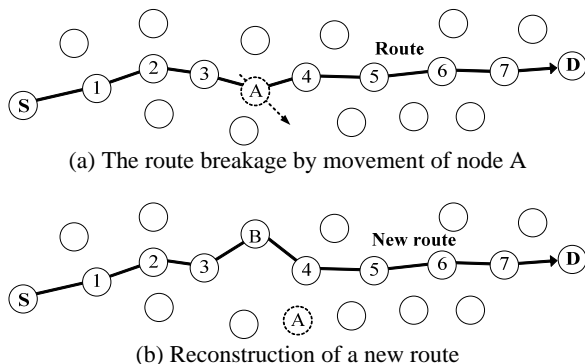


Figure 2. Inefficient route reconstruction by one node

intermediate node moves away as shown in Fig. 2(a). The former node 3 notices the route break to node S and then the route is re-constructed through node B as shown in Fig. 2(b). Comparing Fig. 2(a) and (b), two routes are quite similar. The only difference is that node A is replaced as node B. This example illustrates the worst case of route re-construction. If the route could be recovered locally without notifying the route error to the source, the latency for route rediscovery can be considerably reduced. Second, excessive control packets for route discovery may be generated by route re-constructions. Generally, route discovery process depends on broadcasting route request (RREQ) messages. Each node that receives a RREQ packet re-broadcasts the RREQ message to its neighboring nodes. Uncontrolled broadcasting messages may waste the limited wireless bandwidth and decrease the channel utilization of wireless link. Consequently, the network performance may be deteriorated.

*C. The Limitation of Existing Schemes*

Although existing local repair works try to solve the problem, these schemes have limitations. In LRR [9], the node detecting the route break broadcasts a *route repair* packet with TTL=2. The route repair packet is forwarded to certain nodes which are located two-hops away from detecting node. The reply message then has to undergo same procedures as normal route discovery phase. The whole time required for these processes may take too long. To minimize the bad effect of the route error, the route has to be recovered as soon as it can. In this manner, LRR is not proper for prompt route repair scheme. Also, AODV-BR adopts promiscuous mode to let nodes get explicit information about neighbor nodes. Because adjacent nodes know the situation around the area within transmission range, they can actively participate in local route repair. But continuously operating in promiscuous mode induces much overhead to nodes in terms of energy consumption. Moreover, AODV-BR sometimes recovers the route longer than before. This is another overhead because the source node has to be noticed about changed hop counts. WAR has similar drawbacks. The continuous promiscuous mode operation induces overheads.

IV. QUICK LOCAL REPAIR SCHEME USING ADAPTIVE PROMISCUOUS MODE

The proposed scheme is largely divided into two parts: adaptive promiscuous mode and quick local repair scheme. This section introduces adaptive promiscuous mode and explains an operation of adaptive promiscuous mode. And then, the quick local repair scheme is introduced, which makes use of nodes operating in adaptive promiscuous mode in the next section.

*A. Adaptive Promiscuous Mode*

To achieve the goal of quick route repair, the proposed scheme adopts promiscuous mode in which each node keep monitoring the overheard packets. By the overheard packets, each node can obtain the routing information about the route path in adjacent nodes. Here, the obtained information includes source address, destination

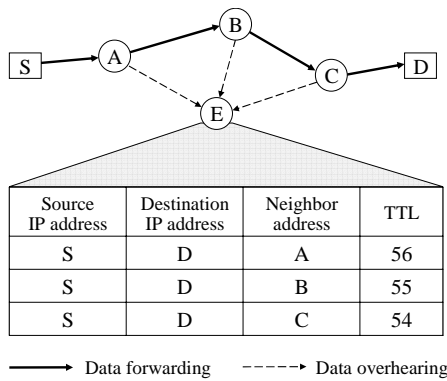


Figure 3. Table structure based on overheard information

addresses, the address of next hop, and etc. The continuous operation in a promiscuous mode can cause excessive energy consumption and reduce network efficiency. However, as explained in the previous section, the routing information obtained from the overheard packets can be utilized for the optimized local repair. So, the proposed adaptive promiscuous mode (APM) tries to take the advantage of promiscuous mode and minimize the overhead of promiscuous mode. Basically, each node repeats promiscuous mode (PM) and non-promiscuous mode (NPM) operation. Only nodes who meet the pre-defined criteria keep promiscuous mode.

1) Table Creation & Management

When nodes operate in promiscuous mode, they read the header of overheard packet. From the packet header, they acquire the information such as where the packet comes from, where it goes to, and how many hops it can be forwarded currently. Based on these informations, nodes create an entry of table which is used for local repair scheme. Nodes overhear packets from all neighbor nodes existing within transmission range. An entry of table includes <source IP address, destination IP address, neighbor address, TTL>, as shown in Fig. 3. For example in Fig. 3, source node S tries to communicate with destination node D, and the route has been constructed as [S-A-B-C-D]. Node E is not a participant of this route. Node E can, however, overhear packets forwarded from node A, B, and C because it exists within their (node A, B, and C) transmission range. After overhearing packets, it creates a table and records each entry based on overheard information. For example, node E records an entry as {S, D, A, 56}. This means that this packet has been transmitted by source node S to destination node D, and forwarded by node A with TTL=56. In case of packets from node B and C, the third item of entry is replaced as B and C. If an entry already exists, i.e. before node has overheard packets from same neighbor, node updates only a fresh item which is usually TTL. Nodes repeat these processes until examining the table fields.

2) The Criteria for Continuous Promiscuous Mode

Adaptive promiscuous mode is devised to utilize promiscuous mode effectively. Our concept is to minimize the time for which nodes operate in promiscuous mode. Hence the criteria are based on

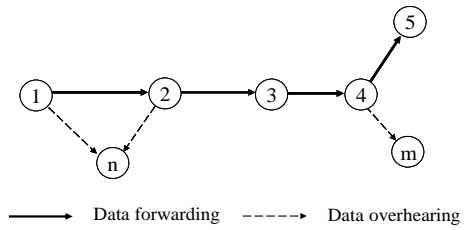


Figure 4. The example of nodes switching to non-promiscuous mode

whether nodes have ability to locally recover the route error.

Consider a topology with seven nodes in Fig. 4. There are five nodes which are participating in an active route and two nodes which operate in promiscuous mode continuously. Packets are forwarded from node 1 to node 5. Node n records two entries through route information obtained from node 1 and 2, and node m has only one entry using packets from node 4. Under these circumstances, assume that one node among node 1, 2, and 4 moves away. Node 3 and 5 are excluded because they have no neighbors. The best way to recover route, when one of node 1, 2, and 4 moves away, is to utilize node n and m. Node n and m, however, do not have enough information around them. In case of node n, it has to know about nodes existing before node 1 or after node 2, i.e. node 3. In former case, node n may replace node 1, and in latter case, node 2. Node m is a similar situation. It has to know about more than two nodes. These nodes like node n and m do not need to operate in promiscuous mode after considering above situation. The criteria for adaptive promiscuous mode makes nodes determine whether they should maintain promiscuous mode. Nodes keep promiscuous mode only if they meet the criteria. The criteria consist of two conditions depending upon the number of adjacent nodes which participate in one routing path (same source-destination IP addresses pair).

- Node should have more than three adjacent nodes in one source-destination IP addresses pair.
- The difference between maximum and minimum TTL has to exceed 2 if node has two adjacent nodes in one source-destination IP addresses pair

Fig. 5 illustrates the examples of the criteria as mentioned above. Fig. 5(a) has three participants and one overhearer. Packets are forwarded from node 1 to node 3

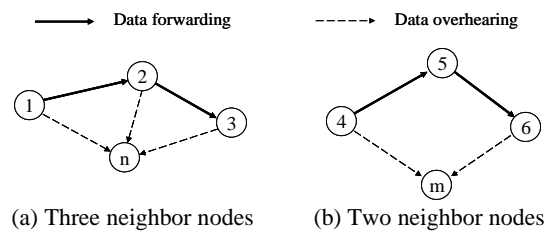


Figure 5. Examples of the criteria for continuous promiscuous mode

via node 2, i.e. [1-2-3]. Node  $n$  is located within the transmission range of these three nodes. Therefore it overhears packets and creates three entries. Under this circumstance, assume that node  $n$  can not function as a part of route, e.g. power down or moving away to another point. The route becomes to break, so it has to be repaired. As illustrated in Fig 5, node  $n$  lies under same conditions comparing with node 2. The transmissions of node 1 and 3 can reach node 2 enough. The hop count from node 1 to node 3 via node 2 is also same as two. In this case, node  $n$  can be a substitute node of the route. After quick local repair process which we propose and will be described in next subsection, the route can be promptly repaired by aid of node  $n$ . Therefore node  $n$  need to maintain promiscuous mode continuously as preparing for route break caused by node 2. All nodes which have more than three neighbors involved in an active routing path have to stay in promiscuous mode. This is the first criteria for continuous promiscuous mode operation.

Fig. 5(b) shows the second criteria. The number of nodes and the nodes' location are similar to Fig. 5(a). It is, however, a little different from Fig. 5(a). Node  $m$  can not notice the transmission of node 5 because it exists out of transmission range of node 5. Node  $m$  only knows about node 4 and 6. It has only two entries, of course. If node  $m$  has two entries about nodes locating adjacently, it can not help any route break as mentioned above. The significant difference is attributed to TTL. In case of node  $m$ , the difference of TTL between two entries is two. It will be one if two neighbors of node  $m$  are adjacent. Node  $m$  perceives that there exists another node which can not be overheard but constitute the route. This is a significant difference. Consequently, the situation of Fig. 5(b) is the same thing as that of Fig. 5(a). Node  $m$  can be a substitute of node 5 as nothing is get changed except routing tables of node 4 and 6. Up to this far, this is the second criteria.

Fig. 6 shows the difference between total number of neighbors and number of neighbors per routing path. Node A and B are not participants of routes, i.e. from S1 to D1 and from S2 to D2. Node A overhears packets from four neighbor nodes. Node B has five neighbors. Although both nodes have more than three neighbor nodes, only node B maintains promiscuous mode. Because node A has actually two neighbor nodes per routing path and difference of TTL is 1, it does not satisfy the requirements based on the criteria. Node B, however, has three neighbor nodes for S1-D1 route. In this way, it will maintain the promiscuous mode.

To sum up, there are two criteria for nodes to maintain promiscuous mode continually. The main point of the criteria is whether nodes can recover the route breaks. To recover the route breaks caused by one node, nodes which operate in promiscuous mode should know about former and latter nodes of node that generates route error among the overall route. Hence overhearing nodes has to acquire information of the route about three nodes in one route or two nodes with different TTL value, 2, in one route.

3) Examining the table

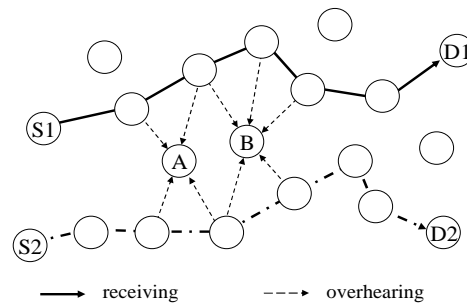


Figure 6. Nodes within multiple routes

With the criteria mentioned above, nodes examine their tables every 't' seconds. We set 't' as 0.5 seconds. After operating in promiscuous mode for 't' seconds, nodes examine their entries in table. If they meet the criteria, promiscuous mode in node goes on. On the other hand, nodes switch promiscuous mode to non-promiscuous mode. They then operate in non-promiscuous mode for 't' seconds. Nodes unconditionally revert to promiscuous mode after non-promiscuous mode. This is to prepare for topology changes. These processes form a set of the operation of adaptive promiscuous mode.

4) Procedures of Adaptive Promiscuous Mode

We assume that nodes start their operation with promiscuous mode. This makes nodes obtain the information about neighbor nodes which act as a part of routing processes. During promiscuous mode, they overhear packets and create a table, if necessary, then update entries based on overheard packets. In case of already existing information, it is discarded without any updates. After 't' seconds, which we set as 0.5 seconds, since nodes operate in promiscuous mode, the table is examined whether they need to maintain the promiscuous mode or not. The determination is done by applying the criteria that have been made to verify the ability of quick local repair. Nodes which meet the criteria keep operating in promiscuous mode, the rest stop the mode. After another 't' seconds, the rest of nodes resume promiscuous mode. Every node repeats above procedures and prepares a sudden route error.

B. Quick Local Repair Scheme

In this section, we introduce the proposed local repair scheme named "quick local repair (QLR)" and explain how it works. The drawback of existing repair schemes, or the long delay, can be solved by aid of overhearing nodes. These nodes mean the ones that exist in the vicinity of the route and currently operate in promiscuous mode. The operation of quick local repair consists of four parts, which are route error detection, sending "HELP" message, examining the possibility of quick local repair, and finishing quick local repair. The details of our scheme will be described with reference to Fig. 7.

Step 1: Route Error Detection

Assume that a source node establishes a route to a destination node in on-demand manner and they communicate with each other. During data transmissions,

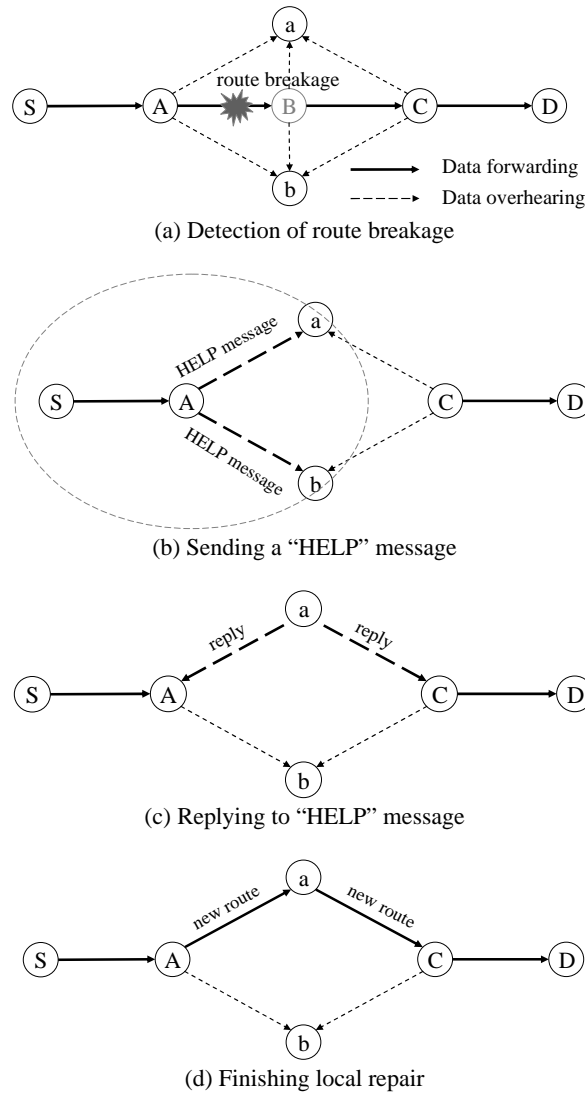


Figure 7. Quick local repair procedures

node B can not act as a part of the route any more because of some reasons, e.g. moving and power off, in Fig. 7(a).

In on-demand routing protocols, nodes periodically checks the connectivity with their former and latter nodes that function as an active route. For example, there is link layer notification. Nodes can be aware of the former & latter nodes' existence when they receive ACK messages after data transmission or CTS messages after sending RTS messages. Nodes also can check the connectivity by passive acknowledgement. This mechanism is activated when link layer notification is not available. In AODV, nodes send "Hello" message to check the connectivity with neighbor nodes. Using above mechanisms, node A notices the link disconnection with node B and executes the quick local repair process.

*Step 2: Sending "HELP" Message*

After noticing a link disconnection, nodes activate the quick local repair process. Primarily, they broadcast "HELP" message aiming for any node which exists within transmission range from them. It is because they do not have any information about how many neighbor nodes exist or which nodes operate in promiscuous mode. The route where the transmission occurs is identified by source & destination IP address. Nodes' own address specifies who detects the route error, and error node's address is for overhearing nodes which have three entries including error node. Fig. 7(b) shows the "HELP" message broadcasting.

*Step 3: Determination of the Possibility of Quick Local Repair*

When overhearing nodes who receive "HELP" messages examine their tables. If they can replace the error node, i.e. they know about latter node of error node, they send approval messages as the answer to "HELP"

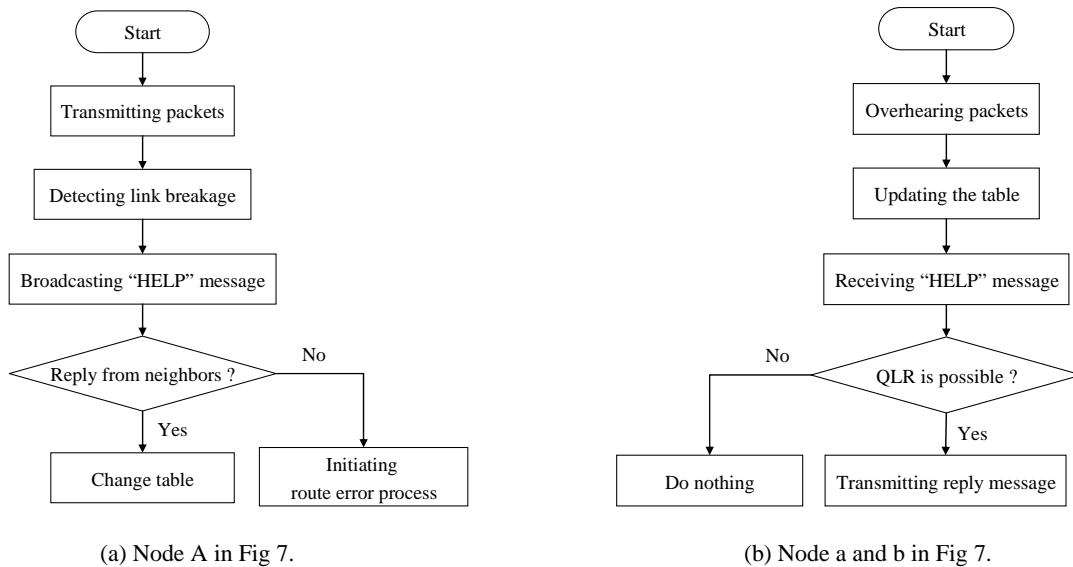


Figure 8. The flow charts of quick local repair

messages. The approval messages are transmitted to former node, which sent “HELP” message, and latter node. When multiple nodes answer to “HELP” message, former node chooses the earliest message. If overhearing nodes do not have enough information to recover route error, they do not answer to the “HELP” messages. In Fig. 7(c), node a sends approval messages reply to HELP message.

*Step 4: Finishing Quick Local Repair*

After receiving the approval message from overhearing node, former and latter node change their routing table. Then, the quick local repair is over. If former nodes do not receive any approval messages, they send a route error message to the source node, and a route may be reconstructed. In Fig. 7(d), a new route is constructed after quick local repair scheme.

*C. Operational Procedures for Each Node*

Fig. 8 shows flow charts of quick local repair. Fig. 8(a) is the flow chart about the procedure of former nodes which detect the route error and send “HELP” message, i.e. node A in Fig. 7. Fig. 8(b) is the flow chart about the procedure of the one of overhearing nodes which replies to “HELP” message, i.e. node a and b in Fig. 7.

V. SIMULATION STUDIES

In this section, we evaluate the performance of our proposed scheme, Quick Local Repair Scheme with Adaptive Promiscuous Mode (QLRS-APM). We utilized AODV as routing protocol and implemented our scheme in AODV. Then we compared the simulated results of our scheme with those of AODV. The simulation results prove that the performances of the proposed scheme outperform those of AODV. Three performance parameters were used to examine the improvement: packet delivery fraction, average end-to-end delay and normalized routing overhead.

*A. Simulation Model*

We have used a detailed simulation model of the NS-2 simulator [13]. There are extensions in NS-2 simulator developed by Monarch research group at Carnegie-Mellon University for supporting simulation of multi-hop wireless networks. Network layer protocols, MAC layer and physical layer models are defined in the extensions. The CMU-extensions have been utilized for our model [14].

*1) Data Link and Physical Layer Models*

The Distributed Coordination Function (DCF) of IEEE 802.11 [12] for wireless LANs is used as a MAC layer protocol. The 802.11 DCF uses Request-To-Send (RTS) and Clear-To-Send (CTS) control packets for unicast data transmission to a neighboring node. Data packet transmissions are followed by hop-by-hop ACK packets. Broadcast data packets and the RTS control packets are sent using physical carrier sensing. An unslotted carrier sense multiple access (CSMA) technique with collision avoidance (CSMA/CA) is used to transmit these packets. The physical radio characteristics of each mobile node’s network interface, such as the antenna gain, transmit power, and receiver sensitivity, were chosen to approximate Lucent WaveLAN with a channel capacity of 2Mbps and a normal radio range of 250 meters. The radio range means the maximum possible distance between two communicating mobile nodes.

The protocol maintains a send buffer with 64 packets. The send buffer contains all data packets waiting for a route, such as packets for which route discovery has started, but no reply has arrived yet. To prevent buffering of packets indefinitely, packets are dropped if they wait in the send buffer for more than 30 sec. all packets (both data and routing control) sent by the routing layer are queued at the interface queue until the MAC layer can transmit them. The interface queue has a maximum size of 50 packets and is maintained as a priority queue with

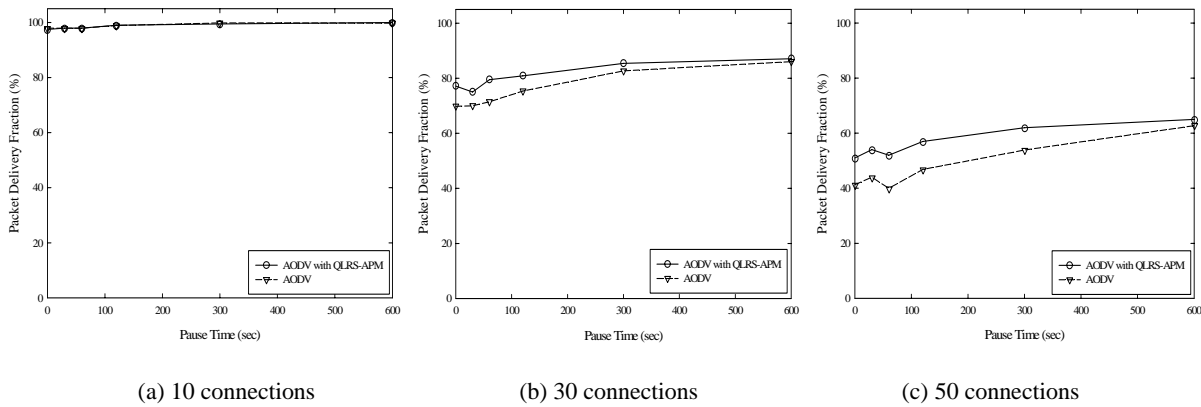


Figure 9. Packet delivery fraction

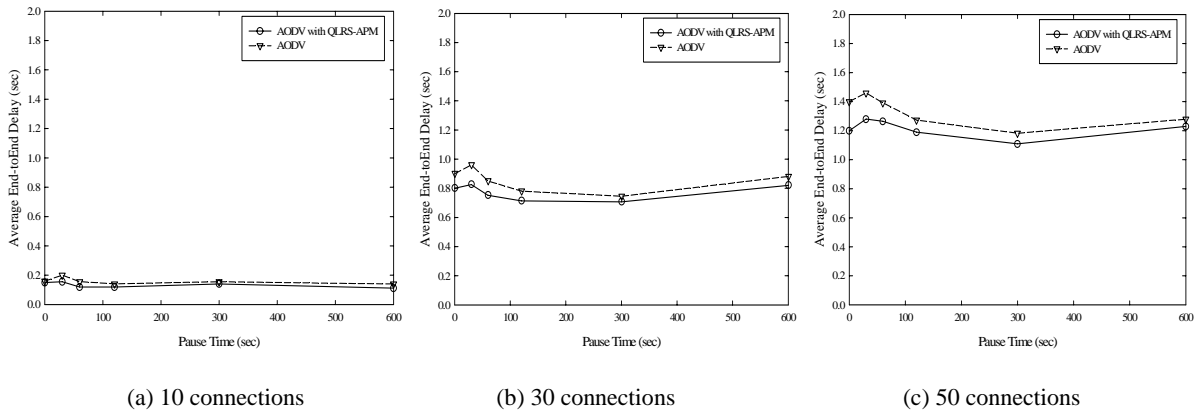


Figure 10. Average end-to-end delay

two priorities each served in FIFO order. Routing packets get higher priority than data packets.

## 2) Traffic and Mobility Models

The mobility model is the random waypoint model in a rectangular field. The movement scenario is characterized by a pause time. Each node starts its movement from a random location to a random destination with a randomly chosen speed (uniformly distributed between 0-20 m/s). When a node reaches the destination, another random destination is chosen in a pause time. Every nodes repeats this behavior until the simulation ends. The pause time indicates the degree of mobility of the ad hoc network. Simulations are run for 600 simulated seconds with 50 nodes in the 1500m X 300m rectangular space. Traffic sources are continuous bit rate (CBR). The source-destination pairs are spread randomly over the network. The CBR packet size is 512 Bytes. The number of source-destination pairs is varied to change the offered load in the network. The simulation parameters are varied as follows. The number of connections (source-destination pairs) is 10, 30, and 50. Pause time between movements of each mobile node: 0, 30, 60, 120, 300, and 600.

## B. Simulation Results

We use AODV as a reference model and modify it to implement the proposed scheme, quick local repair scheme with adaptive promiscuous mode (QLRS-APM). We set several performance metrics to evaluate the performances of QLRs-APM and observe the effects of the local repair or the routing paths. The following four important performance metrics are evaluated:

- Packet delivery fraction: the ratio of the amount of data packets delivered to the destination and total number of data packets sent by source.
- Average end-to-end delay: the average difference between the sending of the data packet and its receipt at the destination. This includes all possible delays caused by route discovery latency, queuing delay at interface queue, propagation and retransmission delays in data link and physical layer.
- Normalized routing overhead: the number of routing packets transmitted per data packet which delivered at the destination.

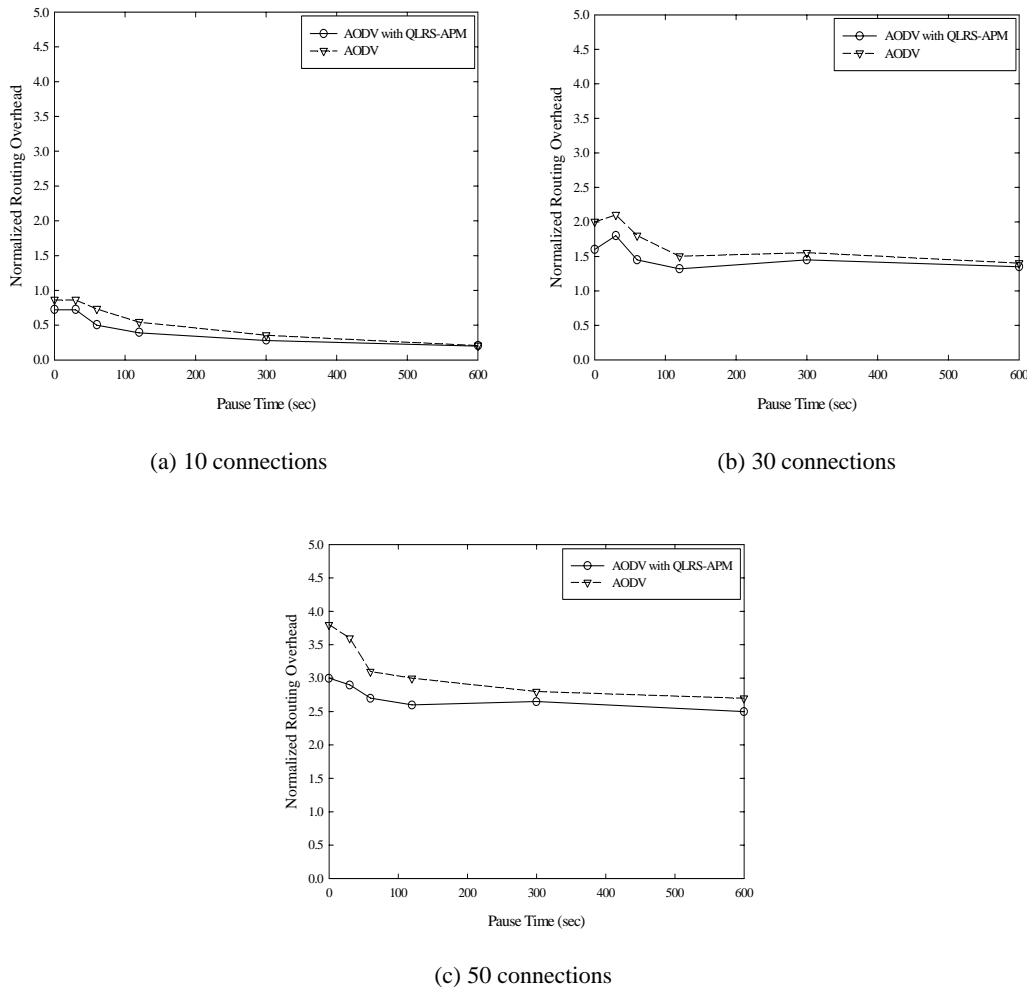


Figure 11. Normalized routing overhead

- Normalized APM overhead: the number of overheard packets at each node during APM compared to those during pure PM. The number of overheard packets during pure PM is set to 1. In case of pure APM in which all nodes don't satisfy the requirements so repeat PM and NPM operation, the value is 1/2.

1) Packet Delivery Fraction

Fig. 9 shows the packet delivery fraction with different number of connections and varying pause times. AODV and AODV with QLRs-APM perform well under a few of connections (light traffic load), but the performance gets worse as the number of connections increases. With the increase of mobility, which means the decrease of pause time, the packet delivery fraction goes down. This is because paths are more likely to break with high mobility. The improvement of QLRs-APM for packet delivery ratio is considerably small in the light load scenario. The reason is that it takes not much time to re-establish the overall path in light load scenario. So, the effect of local repair is small. In higher load scenarios, however, AODV with QLRs-APM shows the higher packet delivery fraction than AODV. And the differences

of packet delivery fraction between AODV and AODV with QLRs-APM get larger as traffic load is higher.

2) Average End-to-End Delay

Fig. 10 shows the average end-to-end delay. The end-to-end delay increases as the number of connections grows since the number of congested nodes along the route increases. The end-to-end delay is also affected by the pause times. At 10 connections, the end-to-end delay of AODV and AODV with QLRs-APM is almost same since there is no congestion and it takes little latency to re-establish a new route. However, at both 30 and 50 connections, the congestions may occur frequently in network. Then, the time for route re-establishment with 30 and 50 connections is much longer than that with 10 connections. So, the average end-to-end delay rises as the number of connections increases. At 30 and 50 connections, the average end-to-end delay unexpectedly rises as the rate of mobility decreases. This result implies that mobility can somehow eliminate congestion.

3) Normalized Routing Overhead

Fig. 11 shows the routing overhead generated throughout the entire simulations with varying the number of connections. Most of routing packets are generated in the route discovery process. Route request

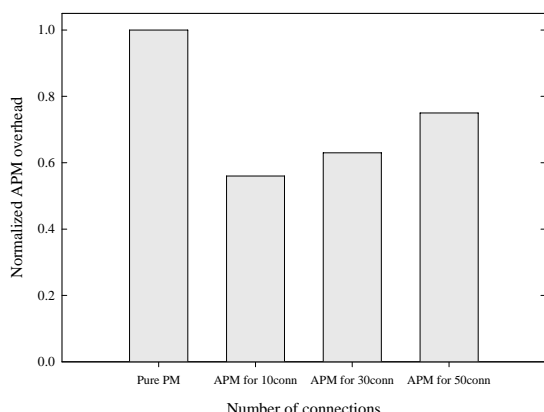


Figure 12. Normalized APM overhead

messages are forwarded to the destination by broadcast. In overall view, the routing overhead increases as the pause time decrease and the traffic load gets higher. In the simulation with 10 connections, both AODV and AODV with QLSR-APM have small routing overhead because the number of source nodes which initiate route discovery process is also small. Then, in this simulation, the effect of quick local repair is not noticeable. However, in the simulations with 30 and 50 connections, source nodes generate many routing messages for re-route discovery processes due to node-mobility. This incurs many routing overheads. As shown in Fig 11(b) and (c). The proposed QLSR-APM then reduces the number of source nodes which start re-route discovery processes. So, the difference of routing overheads goes large in high mobility.

#### 4) Normalized APM Overhead

Fig. 12 shows the normalized APM overhead. This value is to evaluate the effect of APM compared with pure PM (all nodes operate in promiscuous mode continuously). Assume that the overhead of node which continuously operates in pure PM during simulation equals 1. Likewise, if node operates in APM which just repeats PM and NPM, the overhead of this node equals 0.5 compared with pure PM.

As the number of connections increase, nodes overhear packets from several active connections. Therefore they may maintain PM which does not repeating PM and NPM for the quick repair. At 10 connections, normalized APM overhead is almost 0.5. At 30 and 50 connections, however, it almost becomes 0.6 and 0.75. Although the normalized APM overhead is getting larger as the traffic load grows, the operation of APM is still more efficient than the operation of pure PM.

## VI. CONCLUSION

This paper addresses the challenge of providing a quick local repair about route error through the effective usage of promiscuous mode in mobile ad hoc networks. In the proposed schemes, in order to reduce network overhead caused by promiscuous mode in overall time in terms of energy limit, nodes automatically performs the

operation of promiscuous mode only when nodes are able to participate in the repair of a broken route in their local area. Therefore, nodes can minimize resources wasted by unnecessary promiscuous mode. Moreover, using the proposed adaptive promiscuous mode, nodes have the ability to recover the route breakage quickly. Through this scheme, we improve average end-to-end delay through minimizing the repair time spent to recover a broken route and average delivery fraction. Also, we minimize the overhead induced by promiscuous mode.

Simulation show that proposed scheme significantly improves communication performance such as the repair time and the number of packets which are unnecessarily overheard. Our quick local repair scheme with adaptive promiscuous mode can promptly repair a broken route with low overhead. Therefore, the routing protocols implemented our scheme can get the flexibility in ad hoc environments with node-mobility.

## REFERENCES

- [1] D.P. Agrawal, "Future Directions in Mobile Computing," *Mobile Computing and Comm. Rev.*, pp. 13-18, Oct 1999.
- [2] E. M. Royer and C. K. Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks," *IEEE Personal Communications Magazine*, vol. 6, no. 2, pp. 46-55, April 1999.
- [3] D. Johnson et al., "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks Internet Draft)," [www.ietf.org/internet-drafts/draft-ietf-manet-dsr-07.txt](http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-07.txt), Feb 2002.
- [4] C.E. Perkins and P. Bhagwat, "Highly Dynamic Destination Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *Proc. ACM SIGCOMM Symp. Comm., Architectures, and Protocols*, 1994.
- [5] C. E. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing," *Proc. of 2nd IEEE Workshop on Mobile Computing System and Applications*, pp. 90-100, Feb 1999
- [6] D. Johnson and D. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing*, chapter 5, pp. 153-181, Kluwer Academic, 1996
- [7] I. Aron and S. Gupta, "A Witness-Aided Routing Protocol for Mobile Ad-Hoc Networks with Unidirectional Links," *Proc. of International Conference on Mobile Data Access (MDA '99)*, pp. 24-33, Dec. 1999.
- [8] T. Clausen, P. Jacquet, "Optimized Link State Routing Protocol," IETF RFC 3626, <http://ietf.org/rfc/rfc3626.txt>, 2003.
- [9] R. Duggirala, R. Gupta, Q. A. Zeng, and D. P. Agrawal, "Performance Enhancements of Ad Hoc Networks with Localized Route Repair," *IEEE Transactions on Computers*, vol. 52, no. 7, pp. 854-861, Jul 2003.
- [10] S. J. Lee and M. Gerla, "AODV-BR: Backup Routing in Ad hoc Networks," *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 3, pp.1311-1316, Sep 2000.
- [11] G. Liu, K. J. Wong, B. S. Lee, B. C. Seet, C. H. Foh, and L. Zhu, "PATCH: a novel local recovery mechanism for mobile ad-hoc networks," *Proc. of IEEE Vehicular Technology Conference*, vol. 5, pp. 2995-2999, Oct. 2003.
- [12] IEEE, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE Std. 802.11-1997, 1997

- [13] The ns-2 Network Simulator. <http://www.isi.edu/nsnam/ns/>.
- [14] The CMU Monarch Project. <http://www.monarch.cs.emu.edu/>.

**Joo-Sang Youn** is a Ph.D candidate in Dep. of Electronics Engineering at the Korea University. He received the B.S and M.S. degree in Dep. of Electronics Engineering from Korea University in 2001 and 2003, respectively. His current research interests lie in the areas of QoS-aware systems and routing protocol in mobile ad hoc networks and large-scale sensor networks.

**Ji-Hoon Lee** is a senior engineer at Samsung Electronics. He received the B.S., M.S. and Ph.D degree in Dep. of Electronics Engineering from Korea University in 1996, 1998, and 2001 respectively. He is currently working at next generation network group and his current research interests include IEEE 802.11-based wireless mesh networks, large-scale sensor network, and IPv6 over WiBro networks.

**Doo-Hyun Sung** is a research engineer at LG Electronics. He received the B.S. and M.S. degree in Dep. of Electronics Engineering from Korea University in 2004 and 2006 respectively. He is currently working at next generation network group and his current research interests include 3GPP Long Term Evolution and routing protocol in mobile ad hoc networks.

**Chug-Hee Kang** is currently a professor of Dep. of Electronics Engineering at Korea University. He is received B.S., M.S. and Ph.D. degrees from Waseda University, Tokyo, Japan in 1975, 1977 and 1980, respectively. From 1980 to 1994, he was with Electronics and Telecommunications Research Institute (ETRI), Korea, and from 1990 to 1994, he was a vice president of ETRI. In 1994 he was a visiting professor at Washington University, ST. Louis, Missouri. His current research interests lie in the areas of next generation internet.