AODV Multipath Extension using Source Route Lists with Optimized Route Establishment

Yusuke Sakurai and Jiro Katto

Graduate School of Science and Engineering, Waseda University
3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555 Japan.
E-mail: {sakurai, katto}@katto.comm.waseda.ac.jp

***Abstract*— This paper proposes a novel on-demand multipath routing protocol for a mobile ad hoc network. By applying a newly developed route update procedure with combined metrics of delay, hop count and disjointness, each intermediate node deliberately selects multipath candidates while contributing to suppression of unnecessary routing packets. Extension of RREQ/RREP packets with a source route list is also incorporated, not only to alleviate limitation of the hop-count based approaches but rather to provide more efficient multiple routes. Computer simulations using ns-2 simulator are performed with comparison to conventional methods and effectiveness of the proposed methods is quantitatively validated.**

***Index Terms*— Mobile Ad Hoc Network, Multipath Routing, AODV.**

I. INTRODUCTION

A mobile ad hoc network is characterized by its dynamic topology without any stationary infrastructure supports, limited channel bandwidth and limited battery power. Packet transfer is carried out in a multihop manner over wireless links that suffer from radio interference of neighboring nodes. Topological change due to node mobility has to be managed efficiently without causing long link break periods. The number of routing packets has to be reduced as possible as it can be due to limited radio resources. To overcome these problems, a lot of routing protocols had been proposed [1]-[8]. They are categorized into three parts according to the way how their routing tables are constructed: *proactive* [2]-[4], *reactive* (or *on demand*) [5]-[7] and *hybrid* [8]. Detailed studies on their performance comparison are performed [9][10]. As a result, it has been revealed that, in the presence of node mobility, the reactive approaches perform better than the proactive ones. This is mainly due to their low routing overheads. In addition, it has been also shown that, among the reactive approaches, AODV (Ad hoc On-demand Distance Vector) has clear advantages in its moderate overheads and its route convergence performance. Due to these reasons, AODV becomes one of the promising protocols currently available for the mobile ad hoc network.

On the other hand, recent studies extensively focus on multipath extension of the on-demand routing protocols in order to alleviate single-path problems, such as high route discovery latency, frequent route discovery attempts and possible improvement of data transfer throughput. They are categorized into two parts according to the way how the multiple routes are utilized: as backup routes for fault tolerance [11]-[15] and as data transfer routes for load balancing (or path diversity) [16][17]. In the former case, a backup route is activated when the current primary route faces a link failure. In the load balancing case, although this is out of scope of this paper, multiple routes are simultaneously activated and data packets are distributed over them.

Along with the improvement of recent researches on multipath routing, this paper proposes a novel multipath on-demand routing that drastically improves AODV performance and reduces its routing packets. We derive a new route update procedure using combined metrics of delay, hop count and disjointness. According to this procedure, each intermediate node deliberately selects multipath candidates while contributing to suppression of unnecessary routing packets. Extension of RREQ/RREP packets with a source route list is also incorporated. Its purpose is not only to alleviate limitation of the hop-count based approach but also to provide more efficient multiple routes. Computer simulations using ns-2 simulator are performed with comparison to conventional methods. Through various simulations, effectiveness of the proposed methods is quantitatively validated.

II. RELATED WORK

This section summarizes various examples of on-demand multipath routing protocols especially from the viewpoint of route discovery strategy.

AODV Backup Routing (AODV-BR) [11] enhances the AODV by letting each neighboring node of a primary route maintain its own backup route. When the node over a primary route detects a link failure, it transmits a route update message and a neighboring node receiving this message activates the backup route. A problem of this approach is limitation of the route selection that is at most within one hop distance.

MNH (Multiple Next Hops) [12] enables multipath creation without any modification to AODV. In MNH, when an intermediate node receives copies of a RREQ packet, it records multiple reverse routes dissimilar to AODV. When it receives a

RREP packet, it forwards the packet to all the neighboring nodes that are on the reverse routes. RREP packets are finally delivered to a source node and multiple routes are created accordingly. Problems of this method are occurrence of routing loops and production of too many routing packets.

AOMDV (Ad hoc On-demand Multipath Distance Vector routing) [13] is a sophisticated protocol which produces multiple routes with loop-free and link-disjoint properties. When an intermediate node receives copies of a RREQ packet, it compares a hop count field in a packet with the minimum hop count, called *advertised_hopcount*, stored in a routing table for previous RREQ packets. Only a packet with the minimum hop count is accepted to avoid routing loops. Furthermore, a *firsthop* field in a RREQ packet is then compared with the *firsthop_list* in a routing table. When a route with node-disjoint property (new *firsthop*) is found, a new reverse route is recorded. A destination returns RREP packets accordingly, and multiple routes with link-disjoint property are established at a source node. A problem of AOMDV is that several efficient routes may be missed due to strong restriction by the *firsthop* field. Another problem is lack of backup route maintenance that causes timeout expiration of backup routes.

AODVM (AODV Multipath) [14] is a protocol that tries to solve AOMDV's problems above. This method introduces a *jointcount* field into a RREP packet instead of the *firsthop* field and executes bicasting of a RREP packet. This method also introduces a *keep-alive* packet that is periodically inserted to backup routes. These extensions solve AOMDV's problems. However, AODVM still suffers from occurrence of inefficient routes due to limitation of the hop-count based approach.

On the other hand, Multipath DSR [15] is one of DSR extensions that are based on source routing. In this method, a destination node receives multiple RREQ packets and determines multiple routes accordingly. Two kinds of multiple routes, node-disjoint routes and alternate routes branched from a primary route, are considered and it was concluded that the latter performs better. SMR (Split Multipath Routing) [16] is another example of the DSR extensions. In SMR, a destination node receives RREQ packets with hop count limitation and selects maximally disjoint multiple routes. In this case, simultaneous data transfer over the multiple routes is evaluated.

Similarity exists between these DSR extensions and our proposal because a source route list is conveyed by the routing packets. However, a difference lies in the fact that, in our proposal, each intermediate node deliberately updates its route candidates and a source node determines multiple routes finally. In addition, although this is not so important, a source route list is utilized only during route discovery phase, not during data transfer phase.

III. PROPOSED METHODS

A. Overview of AODV

AODV [5] is a single path on-demand routing protocol for a mobile ad-hoc network. It is composed of two phases; route discovery process and route maintenance process, using next three packets.

- ☐ RREQ (Route Request)
- ☐ RREP (Route Reply)
- ☐ RERR (Route Error)

Figure 1 denotes usage of these packets. In route discovery process, a source node firstly broadcasts a RREQ packet towards a destination node. An intermediate node that receives the first RREQ packet records a reverse route to the source node and re-broadcasts the packet. When the intermediate node receives duplicate RREQ packets, it simply discards them. In response to the firstly arrived RREQ packet, the destination node returns a RREP packet to the source node by unicast. An intermediate node that receives the packet records a forward route to the destination and forwards the packet to a neighbor node on the reverse route. The RREP packet finally returns back to the source node and a data transfer route is established.

In route maintenance process, when a node detects a link failure, it generates a RERR packet by broadcast. This RERR packet is propagated over the routes while simultaneously invalidating the corresponding routes. When a RERR packet is sent back to a source node, the source node initiates new route discovery process. In Figure 1, node 7 moves away and a link failure happens between nodes 6 and 7. Then, node 6 detects the link failure and broadcasts a RERR packet.

A problem happens when this route re-discovery process is too frequently initiated due to node mobility or bad channel condition. It causes huge routing overheads and data transfer interruptions, resulting in serious performance degradation.

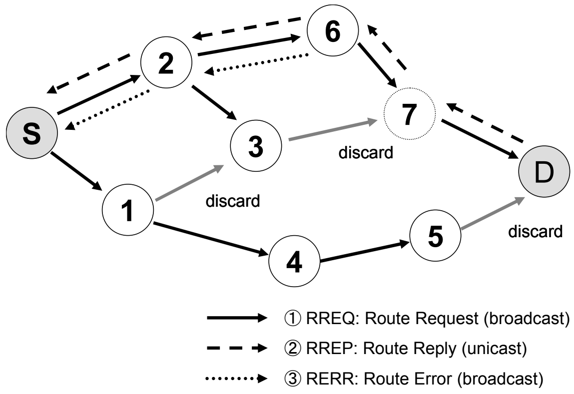


Figure 1: Overview of AODV protocol.

B. Multiple Route Discovery

We extend the route discovery process by letting each intermediate node select reverse routes and forward routes in a distributed manner according to a specified metric. Instead of special fields such as the *firsthop* field of AOMDV and the *jointcount* field of AODVM, a source route list is attached to the RREQ/RREP packets. Figure 2 summarizes our extensions. Details are as follows.

1) RREQ Extension

Similar to AODV, a source node broadcasts a RREQ packet. When an intermediate node receives the first RREQ packet, it records a reverse route in its routing table and re-broadcasts the packet. On the other hand, when the intermediate node receives a delayed RREQ packet from other neighbors, it firstly checks a source route list in the packet and discards the packet when a routing loop is detected. If this address check is passed, the intermediate node then checks reverse routes already stored in its routing table. According to the metric composed of delay, hop count and disjointness, the intermediate node determines acceptance of the delayed RREQ packet and, when accepting it, updates the reverse routes. When we apply a metric of hop count minimization, for example, the reverse route selection is carried out as follows. When the hop count of the duplicate packet is equal to or less than the minimal hop count plus *m* hops (typically zero) in a routing table, the packet is accepted as a new reverse route candidate. When the packet does not satisfy the update condition, it is simply discarded. Irrespective of this decision, the duplicated packet is not re-broadcasted. In practice, the number of reverse routes stored in a routing table is limited to *k*, of which typical value is two. Figure 3 presents an example of routing table extension of our proposal, which is managed by each intermediate node. Route information and an expiration timeout field are stored for each reverse route.

Finally, RREQ packets are delivered to a destination node with different source routes. In Figure 2, an example of the source route lists attached to RREQ packets is presented.

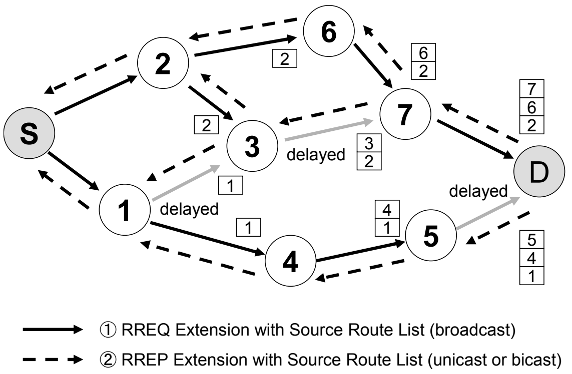


Figure 2: Multiple route discovery by the proposed method.

Destination IP Address	
Sequence Number	
Next Hop 1	route 1 information
Hop Count of Next Hop 1	
Expiration Timeout 1	
Next Hop 2	route 2 information (default : Null)
Hop Count of Next Hop 2	
Expiration Timeout 2	

Figure 3: Routing table extension of the proposed method, when the maximum number of reverse routes is two.

2) *RREP Extension*

A destination node receiving RREQ packets generates multiple RREP packets toward a source node by unicast. The first arriving RREQ packet is unconditionally accepted and a RREP packet is immediately generated to create a primary route, which is usually equal to the AODV route. Delayed RREQ packets are conditionally accepted according to the specified metric. We limit the number of RREP packets to *n*, of which typical value is set to two.

Operation of an intermediate node is slightly complicated. In principle, the intermediate node receiving a RREP packet forwards it to any neighboring nodes over the reverse routes using bicast (multiple unicasts) similar to [14]. However, this bicast causes two problems; (1) occurrence of routing loops, and (2) multiple arrivals of RREP packets. Routing loops can be easily solved by setting the hop-count limitation m≤1 or using an attached source route list. Against the multiple arrivals of RREP packets, we apply a following procedure.

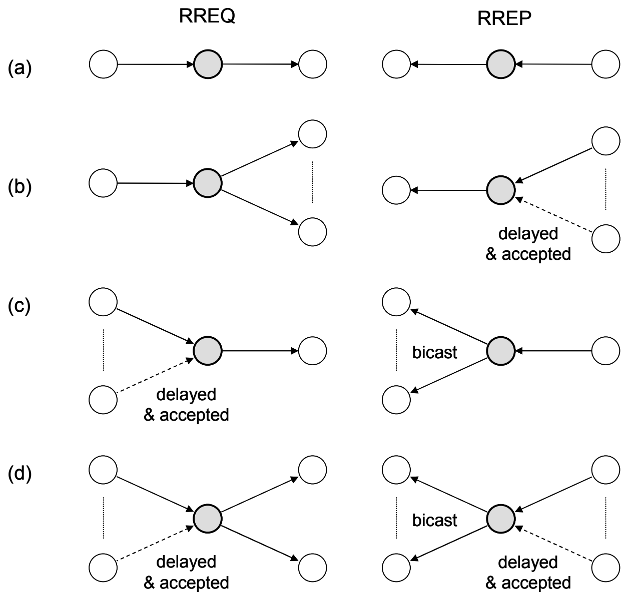


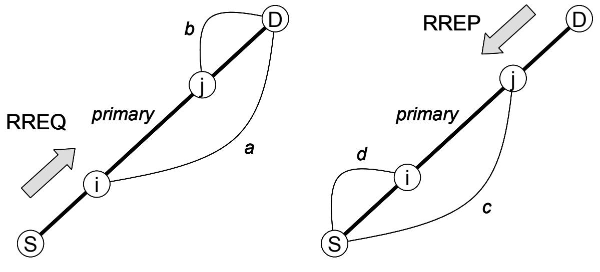
Figure 4: Operations of an intermediate node of the proposed method: (a) basic case similar to AODV, (b) multiple forward routes (delayed update), (c) multiple reverse routes (bicasting), and (d) multiple reverse routes and multiple forward routes (bicasting and delayed update).

When the intermediate node that has multiple reverse routes receives the first RREP packet, it immediately carries out bicasting of the packet. This is valid because reverse route determination in the RREQ process implicitly determines bicasting operation at the intermediate node. When the intermediate node receives a delayed RREP packet, it checks a specified metric condition and decides acceptance of the packet. When the packet is accepted, forward routes are updated in a routing table. Re-bicasting of the packet may be carried out according to the metric definition. Similar to the RREQ process, we limit the number of forward routes to *k*', of which typical value is two. Figure 4 summarizes operations of the intermediate node, which may happen during the RREQ/RREP process.

Finally, RREP packets are delivered to the source node. The fastest RREP packet provides a primary route. Other delayed RREP packets are examined and some of them are accepted as backup routes according to the specified metric. Data transfer begins just after the primary route is established.

C. *Discussion*

We then specify two concrete methods with different metric definitions. The first one is based on a hop count minimization principle (Method 1). Both the reverse routes and the forward routes are updated when delayed RREQ/RREP packets shows less hop counts. In this case, we apply re-unicast or re-bicast to inform the update to a source node. The second method is based on a delay minimization principle (Method 2). Since the metric is delay, RREQ/RREP packets are accepted in their arrival order and no re-unicasting or re-bicasting is performed. We slightly modify the principle that, when a delayed packet shows a hop-count difference larger than m, the packet is not accepted even if the packet arrives fast. This is the result of auxiliary experiments and reflects a background that routes with longer hop counts face a higher probability of link failures [15] even if their estimation shows smaller delays temporally.



(a) (b)

Figure 5: Examples when usage of source route lists contributes to efficient multipath creation: (a) RREQ and (b) RREP.

Furthermore, we combine a source route list with the metrics above to create efficient multiple routes. Figure 5 shows an example of such a case. Figure 5(a) is a case where sub-routes a and b are candidates of an alternate to the primary route. Even if their hop counts are the same, sub-route a is promising because it has higher independency to the primary route and provides robustness against a link failure [15]. Therefore, the destination node selects sub-route a for its RREP return route. Figure 5(b) is the reverse case where the source node selects sub-route c instead of sub-route d because of the route disjointness. Note that this optimization is not possible by the conventional hop-count based approaches such as AOMDV and AODVM.

D. *Multiple Route Maintenance*

When a link failure occurs during data transfer, a RERR packet is broadcasted similar to AODV. When an intermediate node receiving the RERR packet stores an alternate route in its routing table, it switches a route to the alternate route in a localized manner and stops RERR propagation. When a source node receives the RERR packet, it changes a primary route to a backup route or initiates a new route discovery when no backup routes are available. In addition, in order to avoid backup route invalidation and to check the route connectivity, a *keep-alive* packet is inserted periodically into backup routes similar to [14]. In our proposal, the *keep-alive* packet is bicasted when an intermediate node has multiple forward routes.

IV. PERFORMANCE EVALUATION

A. *Simulation Conditions*

We evaluated our proposal using ns-2 simulator [18]. A simulated field is 2200[m] × 600[m] inside which 100 nodes are moving around. Simulations are run for 300 seconds. Each node moves randomly with a speed of 0-20 [m/sec] and stays at the same place with a pause time of 0-300 [sec]. Generally speaking, node mobility is high when the pause time is low and vice versa. The distributed coordination function (DCF) of IEEE 802.11 for wireless LANs is assumed as a MAC layer protocol. Link speed is set to 2Mb/s and a radio range is 250 meters. Ten traffic sources send 512-byte data packets every 0.25 second assuming CBR transmission (16 kbps). Each data point is an average of ten different randomly generated mobility scenarios for the same pause time.

Performance metrics are as follows: (1) packet delivery fraction which is a ratio of the correctly delivered data packets, (2) average end-to-end delay of data transfer, (3) average hop count of the active routes, (4) route discovery frequency, and (5) routing overhead.

B. *Simulation Results*

1) *Packet Delivery Fraction*

Figure 6 shows a comparison of packet delivery fraction. This figure shows that our delay optimized Method 2 performs best among tested methods. This is because our proposal needs the smallest routing packets while providing sufficiently small end-to-end delay and route discovery attempts that are comparable to other methods as below.

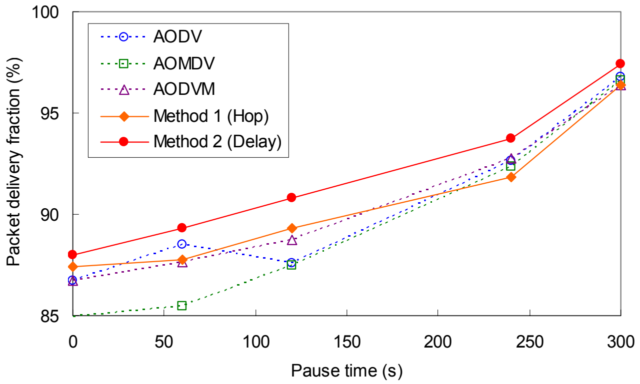


Figure 6: Comparison of packet delivery fraction ratio.

2) *End-to-End Delay*

Figure 7 shows a comparison of end-to-end delays of data packet transfer. This figure shows that multipath extensions provide efficiently small end-to-end delays against AODV. This is because the multipath extensions have backup routes and need smaller route discovery overheads. Another observation is that our Method 2 performs worse when the pause time is 240ms. This is because one simulation scenario resulted in a much bigger delay by chance at that time and needs further study.

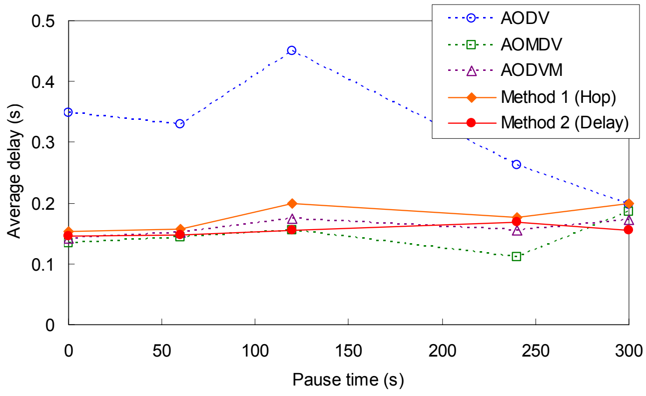


Figure 7: Comparison of end-to-end delay

3) *Hop Count of Active Routes*

Figure 8 shows a comparison of averaged hop counts of the active routes. This figure shows that hop counts are almost similar among tested methods. Also, our hop-count based Method 1 provides the minimum hop distance.

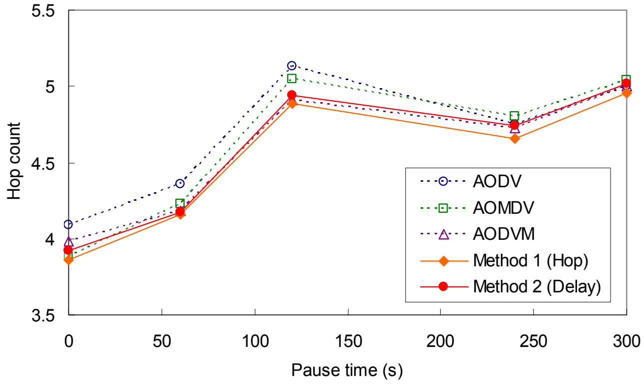


Figure 8: Comparison of hop counts of the active routes.

4) *Route Discovery Frequency*

Figure 9 shows a comparison of routing discovery frequency. This figure supports the previous observation that AODV has to initiate route discovery much more frequently. Method 1 presents the minimum frequency because its smallest hop distance property leads to the smallest chances of link failures.

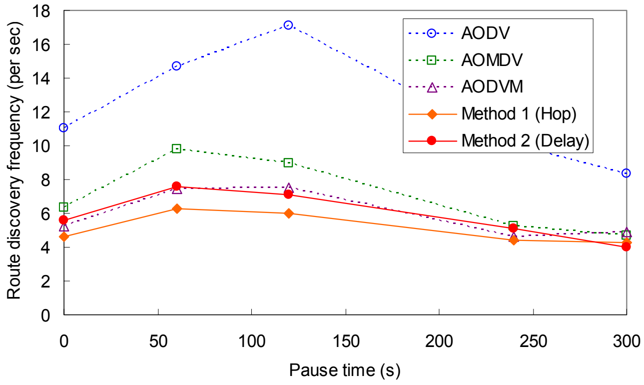


Figure 9: Comparison of route discovery frequency.

5) *Routing Overheads*

Figure 10 shows a comparison of routing overheads when the pause time is zero. This figure shows that our proposals reduce the number of routing packets: 50% decrease against AODV, 25% against AOMDV and 10% against AODVM.

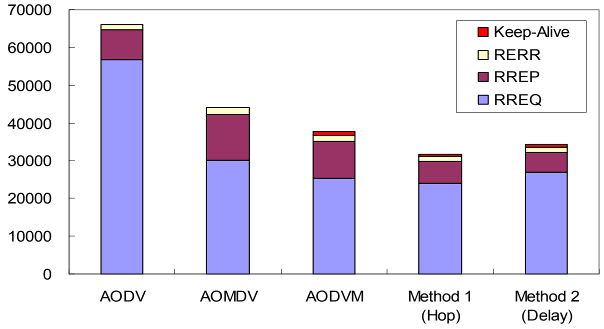


Figure 10: Comparison of routing overheads.

V. CONCLUSIONS

This paper proposed a novel on-demand multipath routing protocol using route optimization with the help of source route lists. Efficient multiple routes were created, which result in higher packet delivery ratio and lower routing packets. Computer simulations were carried out and effectiveness of the proposed methods is validated. As future work, further improvement of the metric definition should be considered.

REFERENCES

[1] C. E. Perkins: "Ad Hoc Networking", Addison Wesley 2000.

[2] C. E. Perkins and P. Bhagwat: "Highly Dynamic Destination-Sequenced Distance-Vector Routing for Mobile Computers", *ACM SIGCOMM '94*, Aug.1994.

[3] A. Laouiti et al: "Multipoint Relaying: An Efficient Technique for Flooding in Mobile Wireless Networks", *IEEE HICSS 2001*, Jan.2001.

[4] T. Clausen and P. Jacquet: "Optimized Link State Routing Protocol", RFC 3626, Oct.2003.

[5] C. E. Perkins et al.: "Ad Hoc On Demand Distance Vector (AODV) Routing", RFC 3561, July.2003.

[6] J. Broch, D. Jhonson, and D. Maltz: "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks", *Mobile Computing*, pp.153-181, 1996.

[7] V. D. Park and M. S. Corson: "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks", *IEEE INFOCOM '97*, Apr.1997.

[8] Z. J. Haas: "A New Routing Protocol for the Reconfigurable Wireless Networks", *IEEE ICUPC '97*, Oct.1997.

[9] J. Broch et al.: "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," *ACM/IEEE MobiCom '98*, Oct.1998.

[10] C. E. Perkins et al: "Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks", *IEEE Personal Communications*, Feb.2001

[11] S. J. Lee and M. Gerla: "AODV-BR: Backup Routing in Ad Hoc Networks", *IEEE WCNC 2000*, Sep.2000.

[12] M. Jiang and R. Jan: "An Efficient Multiple Paths Routing Protocol for Ad-Hoc Networks", *IEEE ICOIN 2001*, Feb.2001.

[13] S. R. Das et al.: "On-demand Multipath Distance Vector Routing for Ad Hoc Networks", *IEEE ICNP 2001*, Nov.2001.

[14] S. Motegi et al.: "Proposal on Multipath Routing for Ad hoc Networks", *IEICE Technical Report*, IN2002-125, Nov.2002 (in Japanese).

[15] A. Nasipuri, R. Castaneda, and S. R. Das: "Performance of Multipath Routing for On-Demand Protocols in Mobile Ad Hoc Networks", *Mobile Networks and Applications*, pp.339-349, Aug.2001.

[16] S. J. Lee and M. Gerla: "Split Multipath Routing with Maximally Disjoint Paths in Ad hoc Networks," *IEEE ICC 2001*, June.2001.

[17] A. Valeram et al: "Cooperative Packet Caching and Shortest Multipath Routing in Mobile Ad hoc Networks," *IEEE INFOCOM 2003*, Apr.2003.

[18] http://www.isi.edu/nsnam/ns/index.html.