Improvement of Throughput Using Partially Node-disjoint Forward and Backward Paths for Mobile Ad Hoc Networks

Mario TAKEUCHI Eitaro KOHNO Tomoyuki OHTA and Yoshiaki KAKUDA
Graduate School of Information Sciences,
Hiroshima City University
3-4-1 Ozuka Higashi, Asaminami-ku, Hiroshima, Japan, 731-3194
Email: {mario@nsw.info., kouno@, ohta@, kakuda@}hiroshima-cu.ac.jp

***Abstract*—In general, many protocols with congestion control utilize acknowledgement packets for the notification of network status. In the case of the communication between a source node and a destination node, data packets and acknowledgement packets share the same path. This can cause packet collisions. AODV is one of the most popular routing protocols of this type for Mobile ad hoc networks. In this paper, we propose a new routing method, which alleviates packet collision using distinct paths for data packets and acknowledgement packets. In this method, we introduce subroute management nodes (SMNs) on the path between the source node and the destination node. We propose a new routing method such that two disjoint paths are established between each two adjacent SMNs. Data packets and acknowledgement packets travel along different paths from each other. This results in alleviation of packet collisions. Through simulation experiments, we show the effect of the packet collision alleviation in comparison with AODV at the number of delivered data packets and acknowledgement packets.**

***Keywords*-Ad hoc networks; AODV; Partially node-disjoint forward and backward paths;**

I. INTRODUCTION

Ad hoc networks [1], which consist of nodes with a routing function, are autonomous distributed networks without any base station. Using intermediate nodes, a node can communicate with another node, even if it can not communicate directly. Many applications for Ad Hoc Networks are being considered. A temporary network after a disaster such as an earthquake is one example. AODV [2][3] is one popular routing protocol.

There are many transport protocols which use data and acknowledgement packets. TCP [4] is the most popular transport protocol, a combination of RTP [5] and RTCP [5] is another example. The reference [6] is a proposal to improve TCP performance by modification of TCP. In that case, transport layer protocols need to know lower layer information. This approach causes many problems such as layer violations and the necessity of transport layer protocols for each datalink layer. Therefore, we attempt to develop a new method without modifying transport protocols. When a source node transfers data packets to a destination node along a path which we call forward path, the destination node sends the packets' acknowledgement to the source node along with the reverse path which we call backward path. In AODV, the forward path and backward path between a source node and a destination node are shared. Therefore, data packets and their return packets can collide. The reference [7] proposed a method which sends the data and the acknowledgement packets along distinct paths using totally node-disjoint paths. However, as a network grows, routes between sources and destinations become longer. In this case, some researchers pointed out that AODV causes overhead, and they proposed countermeasures [8] [9] for that overhead.

In this paper, we propose a new routing method, which utilizes regional loops to alleviate collision. A regional loop consists of two paths between each pair of adjacent SMNs. These two paths are in nodes disjointed from one another. When the routing method forms regional loops, the control packets increase as the hop counts of the loop increase. In the proposed method, we attempt to decrease the amount of control packets using both route splitting and loop forming structure. The proposed method is based on Route-Split Routing (RSR) [9], which is a method for splitting the route and for maintaining it. Nonetheless, RSR is the method for single path routing. We need to extend RSR for forming regional loops. Figure 1 shows an example of route splitting by RSR. In RSR, the Subroute Management Nodes (SMNs) are placed at regular intervals on the path between the source and the destination node. In Figure 1, Node A indicates the source node, and Node J is the destination node. Nodes D and G are SMNs. Our proposed method has been implemented on the simulator, and we compared the throughput, the number of data and acknowledgement packets, and the amount of control packets with AODV.

The rest of the paper is organized as follows: in Section II, we explain our proposed method. In Section III, we mention the simulation experiments and the results. We conclude in Section IV.

II. PROPOSED METHOD

A. Overview of our proposed method

Figure 2 shows an example of paths (loops), which are established by our proposed method. Nodes A and J are the

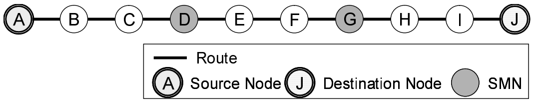


Figure 1. An example network of splitted routes by RSR

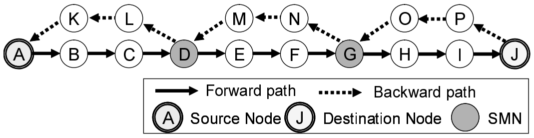


Figure 2. Established routes by the proposed method

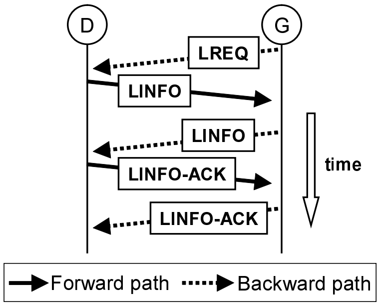


Figure 3. The sequence of the route discovery

source node and the destination node, respectively. Nodes D and G are SMNs. The path A-B-C-D-E-F-G-H-I-J is a forward path and the path J-P-O-G-N-M-D-L-K-A is a backward path.

For convenience, the nearest neighbor to the destination node for an SMN is defined as the next SMN, and the nearest neighbor to the source node for an SMN is defined as the previous SMN. In Figure 2, Node G is the next SMN for node D, and node A is the previous SMN for node D. Our proposed method performs in the same way as other reactive routing methods. Our proposal discovers a route when communication requests on some node occur.

B. Route establishment

In our proposal, the procedures of route establishment are divided as follows:

1) Forward route establishment,
2) Configuring SMNs,
3) Backward route establishment.

We describe each procedure in detail.

1) Forward route establishment: The forward route establishment procedure is divided into forward path discovery and forward path discovery report. To establish a forward route, control messages of RREQ, RREP, and RREP-Ack are introduced.

Forward path discovery: When the communication request occurs on a node (the source node), the source node checks its own routing table. When there is no valid route to the destination node, it will start a procedure for forward route discovery. At first, it broadcasts a RREQ to adjacent nodes. A adjacent node, which receives the RREQ, checks if its own routing table contains a route to the destination address of the RREQ. If this route is not listed, the node adds a route entry to the source node into its own routing table and re-broadcasts the RREQ. This procedure is repeated until the RREQ packets reach the destination node.

Forward path discovery report: When the RREQ reaches the destination node, the destination node sends a RREP to the source node as reply information. The RREP is transferred by unicasting to the source node along the path, which is discovered by the RREQs. An intermediate node, which reveived the RREP, adds a route entry to the source node into its own routing table. In AODV, when a valid route to the destination node already exists in an intermediate node, which received a RREQ, the intermediate node returns a RREP before the destination node does. However, our proposed method does not perform these actions.

2) Configuring SMNs: SMNs are configured when a RREP packet is transferred. When forward route discovery report procedure has been performed, a node, which received the RREP, checks the hop count of the RREP packet. When the node finds that the hop count is equal to multiples of the pre-determined interval, the node becomes a SMN. When the node becomes a SMN, it sends a RREP-Ack to the next SMN by unicasting along the route, which is created by the RREP. The SMN, which received the RREP-Ack, records the address of the generator node as the address of the previous SMN. As the actions are repeated, each SMN address is configured between the source node and the destination node.

3) Backward route establishment: For the backward route establishment procedure, the following three messages are introduced:

- LREQ (Loop REQuest),
- LINFO (Loop INFOrmation),
- LINFO-Ack (Loop INFOrmation Acknowledgement).

We explain the backward path establishment between nodes D and G used in Figure 2. Figure 3 depicts a sequence of the backward path establishment procedure.

Backward path discovery: Node G, which received a RREP-Ack, broadcasts a LREQ toward the node D. A node, which received the LREQ, checks its own routing table. When its own node address is included in the forward path, the node discards the received LREQ. This process ensures that the nodes, except SMNs, will be on either the forward path or backward path. When the address of the node is not included in the forward path, the node adds its own address into the LREQ and re-broadcasts the updated LREQ.

Backward path discovery report: When the LREQ reaches node D, node D generates a LINFO. At the same

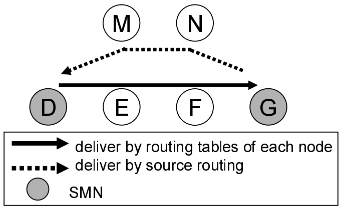


Figure 4. The flow of LINFO

time, it appends addresses obtained from the LREQ (that is, addresses of N and M) to the LINFO. Figure 4 illustrates the flow of the LINFO. Node D sends the LINFO by unicasting to node G along the forward path. Node G, which received the LINFO, refers to addresses within the LINFO, and it sends the LINFO by source routing to the nodes along the backward path. Nodes N and M, which received the LINFO, append the route toward the source node and the previous SMN, node D, to their own routing tables. When the LINFO reaches node D, it begins to perform the backward path availablility report procedure.

Backward path availability report: Node D sends a LINFO-Ack to node G using unicast. Node G updates the next hop toward the source node and node D to node N in its own routing table. This is the start of using the backward path. At the same time, node G sends the LINFO-Ack to node D along the backward path. When node D receives the LINFO-Ack, the procedure finishes with the establishment of the backward path.

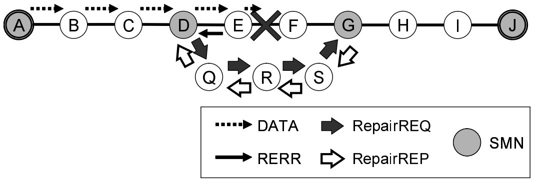
C. Route maintenance

When established paths break due to movements of the nodes, the nodes start route maintenance procedures. As we mentioned before, the procedures of route maintenance are performed between each two adjacent SMNs. In our proposed method, the loop consists of forward and backward paths, so the procedures are different depending on which path is broken. The route maintenance performs the following procedures:

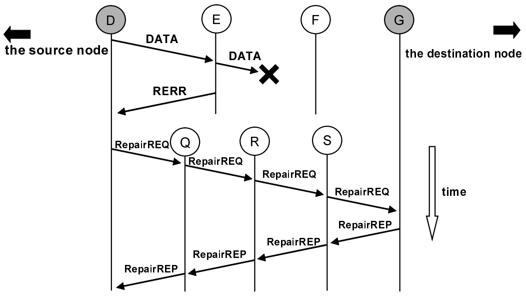
- Forward route maintenance,
  1) Route breakage report,
  2) Route discovery,
  3) Route discovery report,
- Backward route maintenance,
  1) Route breakage detection,
  2) Route breakage report,
  3) Route recovery.

The following control messages are used for route maintenance:

- RERR (Route ERRor),
- RepairREQ (Repair REQuest),
- RepairRREP (Repair REPly),
- KeepAlive.



(a) The network status



(b) The sequence

Figure 5. An example of forward route recovery

Among these, RERR, RepairREQ, and RepairREP are used for route recovery. We explain the procedure of route recovery using the example in Figure 5. In Figure 5, assume that nodes A, D, G, and J are SMNs and the link between nodes E and F is broken.

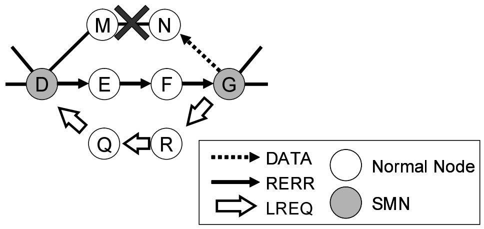
1) Forward route maintenance:

Route breakage report: When node E detects a link breakage to the next hop, node F, it sends a RERR to the previous SMN, node D, for notification of the link breakage. When node D receives the RERR, it suspends sending data packets to the destination node. The data packets will be stored in the buffers of node D which will resume sending after the maintenance procedure is completed.

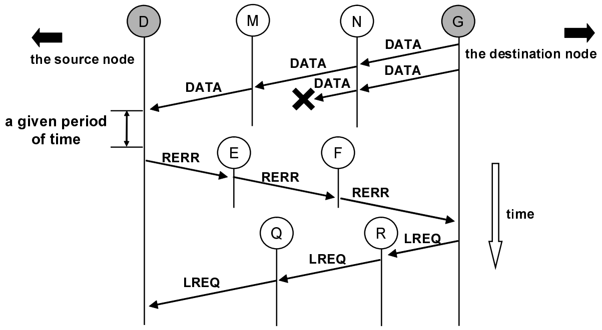
Route discovery: Node D, which receives the RERR, broadcasts a RepairREQ to start the route discovery. At the same time, the node calculates TTL (Time To Live) as the value to the hop counts to node G plus one. This TTL prevents exponential escalation of the RepairREQ. Each node, which received the RepairREQ, confirms whether its own address is included in the backward route. When its address is included in the backward route, the node discards the RepairREQ. Otherwise, the node appends the path to node D to its own routing table, and re-broadcasts the RepairREQ.

Route discovery report: When node G receives the RepairREQ, it sends a RepairREP to node D by unicast. The RepairREP is transferred along the path, which is discovered by the RepairREQs. When the RepairREP reaches node D, the procedure of route maintenance is completed. Then, data packets can continue to be transfered.

2) Backward path recovery: We will explain the procedure of backward path recovery using the example in Figure 6. In Figure 6, assume that nodes D and G are SMNs



(a) The network status



(b) The sequence

Figure 6. An example of backward route recovery

and the link between nodes N and M is broken.

Route breakage detection: SMN D judges that the backward path is broken when it does not receive any packets from the next SMN G during a given period of time.

Route breakage report: SMN D, which detects breakage of the backward path, generates a RERR. Then, it unicasts the RERR to the next SMN, node G, and reports the breakage of the backward path.

Route recovery: The next SMN, node G, which receives the RERR, suspends transferring acknowledgement packets for data, and stores them in its own buffer. Then, node G broadcasts a LREQ to the previous SMN, node D, and it recovers the backward path. When maintenance procedures are completed, it restarts transferring acknowledgement packets for data.

3) KeepAlive: Figure 7 shows the flow of KeepAlive and indicates the breakage of the backward path between nodes D and G. Nodes A, D, and G are SMNs. While node G recovers the backward path, it suspends transferring packets. Sometimes this causes an interruption of any packets on the backward path between nodes D and A. When node A detects interruption of packets during a given time, it assumes that the backward route is disconnected and starts the unnecessary backward route recovery processes. KeepAlive messages are used for the prevention of this side-effect. SMN D, which has not sent any packets to previous SMN A during a given time, sends a KeepAlive to SMN A.

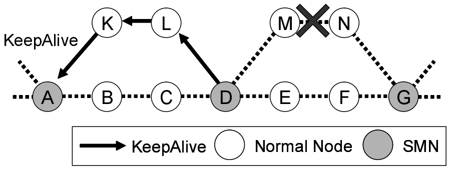


Figure 7. The flow of KeepAlive

Table I
PARAMETERS FOR SIMULATIONS

Simulator	QualNet ver.4.5 [12]
Field size [m^2]	3500×3500
Number of nodes	500
Number of source and destination nodes pairs	10
Number of packets to send	1000
Mobility model	Random Waypoint Model [10]
Pause time [sec]	0
Maximum node speed [m/sec]	1, 5, 10
Data packet size [byte]	512
Ack packet size [byte]	16
Interval time of packets [sec]	0.25
Transmission Range [m]	250
Radio Propagation Path Loss Model	Two-Ray Model
Bandwidth [Mbps]	11
Duration time for simulation [sec]	1300

4) Route re-establishment: When a SMN fails to recover a path between some two adjacent SMNs, it starts the procedures for re-establishing the route. The SMN which fails to recover the route to another SMN, generates a RERR, and sends it to the source node. The source node, which receives the RERR, broadcasts a RREQ, and re-establishes the route to the destination node.

III. SIMULATION EXPERIMENT

We evaluated our proposed method from the viewpoints of the throughput, the number of delivered data packets and the amount of control packets in comparison with AODV.

A. Simulation environment

Table I shows the parameters of simulation experiments. We set the pause time zero for the Random Waypoint Model [10]. Also, we use IEEE 802.11b [11] for the MAC layer protocol.

B. Simulation method

We performed simulations with varying node speeds in the proposed method and AODV. The iteration of simulations is ten times for each node speed. The following simulation is performed; 1,030 seconds from the start of the simulation, each source node starts to send data packets to the destination node at 0.25 second intervals. When the destination node receives one data packet, it returns one acknowledgement packet to the source node. Therefore, the number of data and acknowledgement packets are equal without any packet losses. However, the size of each packet of data and

Table II
THE COMMON PARAMETERS FOR THE PROPOSED METHOD AND AODV

TTL-START	1
TTL-INCREMENT	2
TTL-THRESHOLD	7
NET-DIAMETER	35
RREQ-RETRIES	7
ACTIVE-ROUTE-TIMEOUT [sec]	3
BUFFER-MAX-PACKET	100

acknowledgement is different as can be seen in table I; in other words the amount of data and acknowledgement packets are different. In the experiments of our proposed method, the hop interval between each two adjacent SMNs is set at 2, 5, and 30. In AODV, we set an available route maintenance function. Table II shows the summarization of the common parameters for our proposed method and AODV.

C. Results

Figure 8 shows the throughput of our proposed method and AODV. The horizontal axis indicates maximum node speeds and the vertical axis shows the throughput. Our proposal at every interval between SMNs has higher throughput than that of AODV. At every intervals of our proposal, although all variations had a decrease throughput, the shorter intervals had greater rate of decrease. At interval 30, our proposal maintain a higher throughput. Figures 9(a) and 9(b) show the number of data and acknowledgement packets, respectively. The horizontal axis indicates maximum node speeds and the vertical axis shows the number of data or acknowledgement packets. Our proposal has the higher number of data/acknowledgement packets than AODVs. Also, at every interval of our proposal, the longer intervals have a greater number of data/acknowledgement packets. That shows that the augmentation of the number of received data packets affects the improvement of our proposal in Figure 8.

Next, Figure 10 shows the average number of hop counts. Figure 10(a) and 10(b) are the average number of forward and backward paths hop counts, respectively. Our proposal has longer hop counts than that of AODV. In addition, our proposal which was shorter SMN intervals has tendency the longer hop counts. In our proposal, SMNs divide the paths between the source and the destination.Therefore, the paths between the source and the destination are not the shortest ones. Also, in our proposal, the two paths are distinct from each other. Thus, the two paths are not the shortest paths. Finally, our proposal shows that the average hop counts are longer than AODV.

Figure 11 indicates the amount of the control packets. Figures 11(a), 11(b), and 11(c) show the results of the maximum node speeds at 1, 5, and 10 [m/s], respectively. Our proposal has a greater amount of packets than AODV. Figures 11(a) and 11(b) show that the smallest control packets can be found at intevals 1 and 5, and it is greater 8∼22% control packets than AODV. In the case of SMN intervals 2 and 30, the number increase to 41∼94% greater than AODV. From Figure 11(c), when the node speed is 10 [m/s], SMN intervals 2 and 5 have the almost the same amount of control packets, and that is 42∼46% greater than that of AODV. Also, at interval 30, the number of control packets is 127% greater than that of AODV.

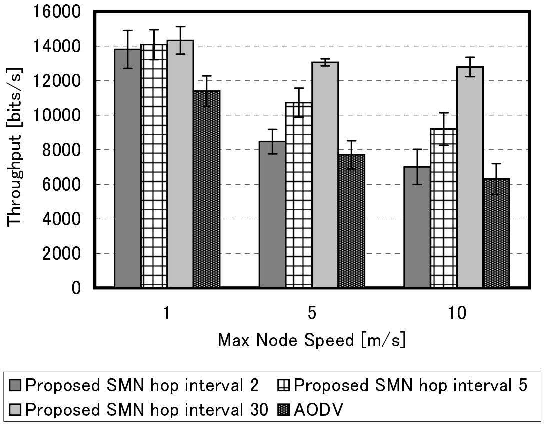


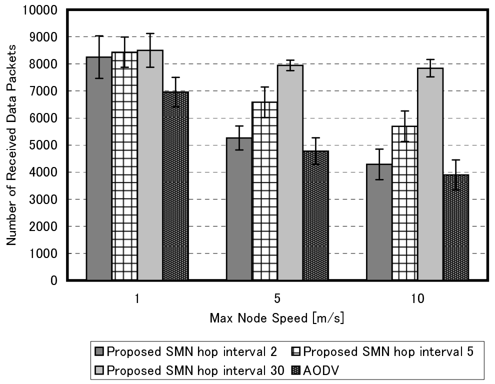
Figure 8. Throughput for AODV versus our protocols

D. Experiment of MAC overhead

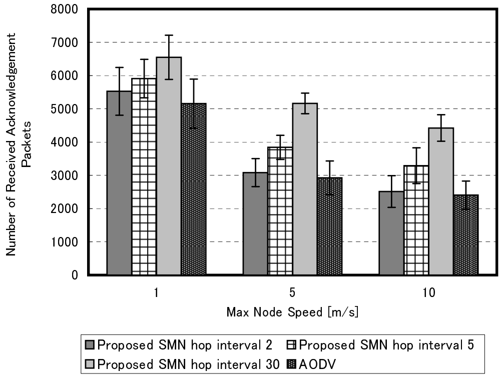
To confirm the effect of alleviation of packet collisions, we have observed the behavior of the MAC layer protocol packets.

Regarding the specification of IEEE802.11b [13] [11] , when the node can not receive an acknowledgement frame after sending a data frame, it sets a back-off time and retransmits the frame. We compared the number of back-offs between our proposal and AODV. Figure 12 shows the summary of the number of back-offs on all nodes. Figures 12(a), 12(b), and 12(c) are the results when the node speed is 1, 5, and 10 [m/s], respectively. The number of back-offs are almost the same as our proposal at SMN interval 5 and AODV. Also, the number of back-offs are almost the same as our proposal at SMN interval 2 when the node speed is 10 [m/s] and AODV. In the case of SMN interval 30 our proposal has a greater number of back-offs than that of AODV.

In addition, we investigate the distribution of the number of nodes versus the total number of back-offs of IEEE802.11b. Regarding IEEE802.11b, when the source node cannot receive an acknowledgement frame, it tries to re-transmit a frame up to seven times. When the source node can not receive an acknowledgement frame after transmitting the seventh frame, it discards the frame. In ad hoc networks, when a node discards a frame after the seventh transmission, it is highly possibile that the route between the source and

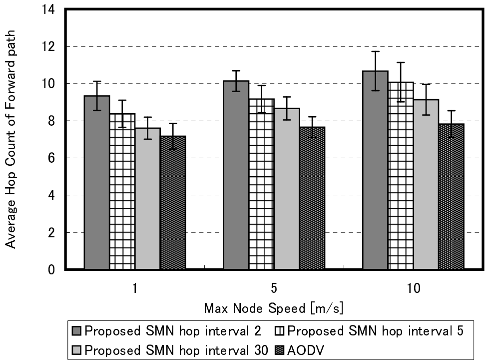


(a) Number of data packets

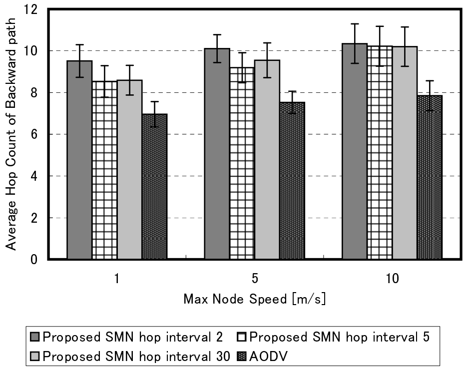


(b) Number of acknowledgement packets

Figure 9. The number of delivered packets



(a) Average hop counts of forward path



(b) Average hop counts of bacward path

Figure 10. Average hop counts for the forward and the backward paths

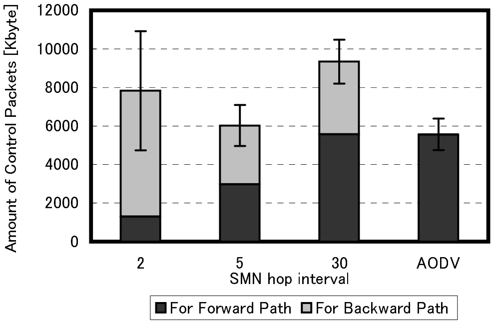
the destination is broken. Therefore we eliminate the number of back-offs at the seventh transmission. Figure 13 shows the distribution of the nodes versus the total number of back-offs. Figures 13(a), 13(b), and 13(c) are the results at maximum node speeds 1, 5, and 10 [m/s], respectively. When the SMN intervals are 2 and 5, the peak value of our proposal is shifted toward the lower number of back-offs. Conversely, the results show that when the SMN interval is at 30, the peak shifts toward the higher number of back-offs.

E. Discussion

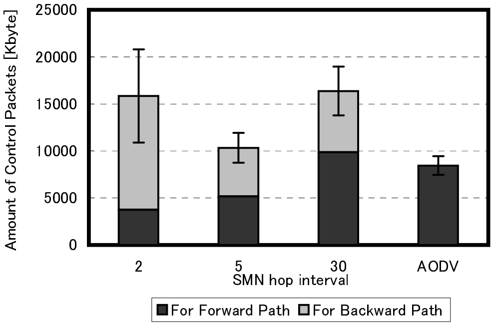
Our proposed method differs from AODV in the following ways:

- It maintains routes within a regional area.
- It introduces loops which consist of a forward and a backward path.

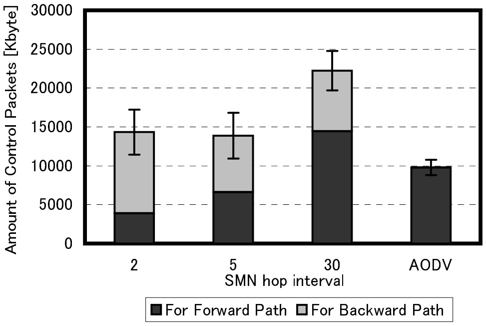
1) Regional route maintenance using route splitting: In mobile ad hoc networks, an increase in the amount of control packets on routing protocols consumes available bandwidth. Many transport protocols like TCP, for example, perform aggressive congestion control. When those transport protocols are used, the bursty background traffic may lead to degradation in the performance of the network. It shows that the sliding-window is introduced to improve the scalability of the protocol on high-bandwidth networks. According to reference [14], when TCP is used, the use of sliding-windows causes throughput degradation in ad hoc wireless networks. Also, the use of a sliding-window may degrade performance in bandwidth-constrained ad hoc wireless networks where the MAC layer protocol may not exhibit short-term and long-term fairness. Consequently, it is important to reduce the amount of control packets. Regional route maintenance using route splitting prevents the escalation of control packets which is caused by route breakage. Figures 11(a), 11(b) show that SMN interval 5 has the smallest amount of control packets. This is the effect of route splitting. In the case of SMN interval 2, the amount of control packets is 44∼88% greater than that of AODV. When hop distances between SMNs are too short, it is difficult to establish two node-disjoint paths. This cause the escalation of the amount of control packets. In the case of SMN interval 30, it almost never uses route splitting by SMNs. As a result, our proposal can not reduce the amount of control packets effectively. Therefore, interval 30 has the greatest amount of control packets. Figure 11(c) shows that the amount of control packets at SMN intervals 2 and 5 are almost the



(a) Maximum Node Speed 1 (m/s)

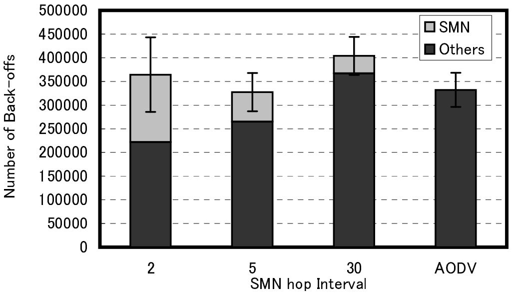


(b) Maximum Node Speed 5 (m/s)

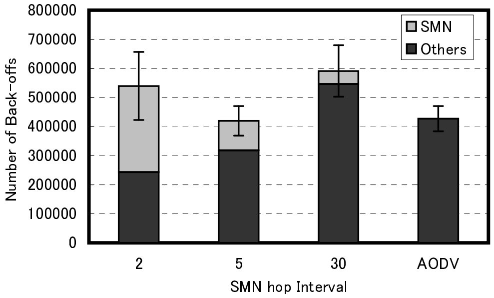


(c) Maximum Node Speed 10 (m/s)

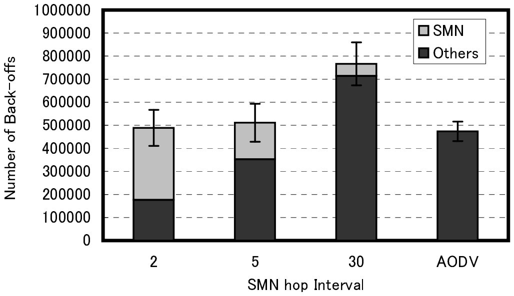
Figure 11. The amount of control packets



(a) Maximum Node Speed 1 (m/s)



(b) Maximum Node Speed 5 (m/s)



(c) Maximum Node Speed 10 (m/s)

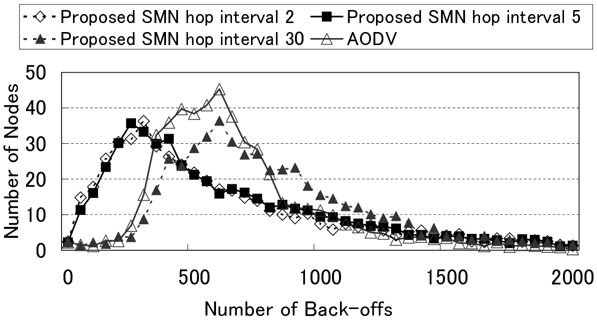
Figure 12. The number of back-off retransmissions of data packets

same. It is because when the maximum node speed is high, it is easy to establish two node-disjoint paths. In the case of SMN interval 30, the amount of control packets is 127% greater than that of AODV. It is for when the maximum node speeds are 1 and 5 [m/s].

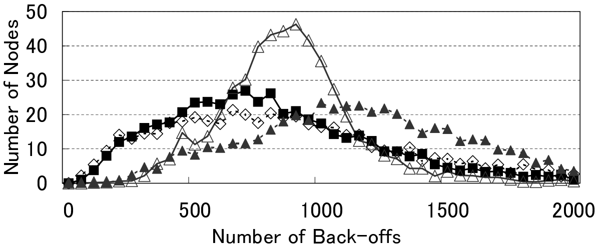
2) Loops, which consist of a forward and backward path: The number of received data packets decreases with an increase in the collisions of packets. In our proposal, we introduced the partially node-disjoint forward and backward paths and attempted to alleviate this phenomenon. Figure 12 shows the number of back-offs may be greater for our proposals. However, Figure 13 shows the peak values of our proposal are shifted toward the lower number of back-offs, when the SMN intervals are 2 and 5. Therefore, it would appear that the number of back-offs which are concentrated on adjacent nodes, is smaller than that of AODV. Also, in the case of the SMN interval 30, the peak value is shifted toward the higher number of back-offs. It is for this reason that the number of received data packets, the amount of control packets and the number of sent packets per node is greater than AODV and our proposal at SMN interval 2 and 5.

IV. CONCLUSION

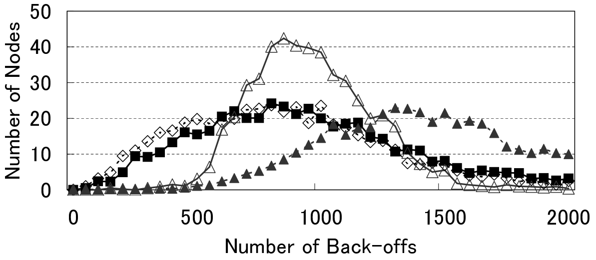
In this paper, we proposed a new routing method which alleviates the collision of packets using partially node-disjoint forward and reverse paths. To confirm the effectiveness of our proposed method, we performed simulation experiments. We compared our proposed method with AODV at the throughput, the number of received data/acknowledgement packets, the number of control packets, and the number of back-offs on IEEE802.11b. As a result, our re-



(a) Maximum Node Speed 1 (m/s)



(b) Maximum Node Speed 5 (m/s)



(c) Maximum Node Speed 10 (m/s)

Figure 13. The distribution of nodes to total number of back-off retransmissions on IEEE802.11b

sults show our proposal has a higher number of received data/acknowledgement packets, and a higher throughput. Also, due to the placement of SMNs, our proposed method can suppress the number of control packets by adjusting the intervals between SMNs. In addition, we can confirme the alleviation between the data and the acknowledgement packets by evaluating the number of back-offs under IEEE802.11b.

In the future, we plan to analyze differences by comparison between our proposal and other multipath protocols in detail. Additionally, we should investigate influence of simulation parameters.

ACKNOWLEDGEMENT

This work was supported in part by the Ministry of Education, Science, Sports and Culture of Japan under Grant-in-Aid for Scientific Research (B) (No.21300028).

REFERENCES

[1] C.-K. Toh, *Ad Hoc Mobile Wireless Networks Protocols and Systems*. Prentice Hall Inc., Dec. 2002.

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

[3] C. Perkins, E. Belding-Royer, and S. Das, “Ad hoc on-demand distance vector (AODV) routing,” RFC 3561 (Experimental), Internet Engineering Task Force, Jul. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3561.txt

[4] J. Postel, “Transmission control protocol,” RFC 793 (Standard), Internet Engineering Task Force, Sep. 1981. [Online]. Available: http://www.ietf.org/rfc/rfc793.txt

[5] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A transport protocol for real-time applications,” RFC 3550 (Standard), Internet Engineering Task Force, Jul. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3550.txt

[6] K. Rojviboonchai, T. Osuga, and H. Aida, “Delay time-based data transmission sequence for multipath transmission control protocol (M/TCP),” *Communications, Computers and signal Processing, IEEE*, vol. 2, pp. 976– 979, 2003.

[7] C. Liu, S. Goto, T. Ikenaga, and T. Baba, “Improving the TCP performance to support mobile multimedia application,” *The 2nd International Conference on Advances in Mobile Multimedia (MoMM2004)*, Sep. 2004.

[8] R. Bai and M. Singhal, “DOA: DSR over AODV routing for mobile ad hoc networks,” *IEEE Transactions on Mobile Computing*, vol. 5, no. 10, pp. 1403–1416, Oct. 2006.

[9] T. Mizumoto, T. Ohta, and Y. Kakuda, “Route-split routing resilient to simulations failure for mobile ad hoc networks,” *IEICE Transaction on Fundamentals*, vol. E91-A, no. 7, pp. 1625–1633, Jul. 2008.

[10] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, “A performance comparison of multi-hop wireless ad hoc network routing protocols,” in *MobiCom ’98: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*. New York, NY, USA: ACM, Oct. 1998, pp. 85–97.

[11] *Supplement to IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements – Part11:Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High-Speed Physical Layer Extension in the 2.4GHz Band*, IEEE Std. 802.11b-1999 (R2003), Sep. 2000.

[12] Scalable Network Technologies Inc., “Qualnet network simulator by scalable network technologies.” [Online]. Available: http://www.scalable-networks.com/

[13] S. Xu and T. Saadawi, “Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks?” *IEEE Communication Magazine*, vol. 39, no. 6, pp. 130–137, Jun. 2001.

[14] C. S. R. Murthy and B. S. Manoj, *Ad Hoc Wireless Networks: Architectures and Protocols*. Prentice Hall Inc., Jun. 2004.