Proposal of autonomous transmission timing control scheme for collision avoidance in ad hoc multicasting

Katsuhiro Naito†, Yasunori Fukuda‡, Kazuo Mori†, and Hideo Kobayashi†

†Department of Electrical and Electronic Engineering, Mie University
1577 Kurimamachiya, Tsu, Mie, 514-8507, Japan, {naito, kmori, koba}@elec.mie-u.ac.jp
‡OMRON Corporation
Shiokoji Horikawa, Shimogyo-ku, Kyoto, 600-8530, Japan, yasunori_fukuda@omron.co.jp

ABSTRACT

In this paper, we propose an autonomous transmission timing control scheme for collision avoidance in ad hoc multicasting. In ad hoc multicasting, packet collisions due to hidden node problems at upstream nodes cause packet losses at all downstream nodes. Therefore, collision avoidance mechanisms are important to improve packet delivery ratio at destination nodes. In this paper, we extend an On-Demand Multicast Routing Protocol (ODMRP) as a base multicast routing protocol to implement transmission timing control mechanisms. In the proposed protocol, each node constructs a neighbor-node list, and informs the neighbor node list to its own upstream node. As a result, upstream nodes can detect their hidden nodes by using these neighbor-node lists from their downstream nodes. Then, each node can select different transmission timing autonomously not to conflict the transmission timing. From the simulation results, it is shown that the proposed method can achieve high data delivery ratio by reducing packet corruptions.

Keywords: Ad hoc networks, Multicast, Routing, Hidden nodes, Transmission timing control

1 Introduction

An ad hoc network is a dynamic autonomous wireless network formed by mobile devices with wireless communication capability. Due to a limited radio propagation range of wireless devices, each node behaves as a router as well as an end host. Therefore, end-to-end communication is performed by multi-hop communication. In order to achieve the multi-hop communication, each node requires a routing protocol for route construction. Several routing protocols have been proposed in recent years for ad hoc networks [1].

Various applications are focused in ad hoc networks. Multicast communication is required for new type applications such as a video streaming. Multicasting in ad hoc networks faces many challenges due to changes in network topology and features in wireless communication environment. Therefore, conventional routing protocols for wired networks cannot apply to ad hoc networks. Some multicast routing protocols have been newly proposed for ad hoc networks [2], [3].

IEEE 802.11 is a candidate device for forming ad hoc networks [4]. Generally, unicast mode communication can avoid collisions due to hidden node problems by using RTS (Request To Send) / CTS (Clear To Send) mechanisms. Meanwhile, broadcast mode communication is used for multicasting in the IEEE 802.11 systems. Since a sender node only performs channel sensing in the broadcast mode communication, it is difficult to avoid collisions due to hidden node problems [5]–[7]. Moreover, some nodes transmit same data packets at same timing in multicasting. Therefore, the hidden node problems cause many packet losses due to collisions. Additionally, almost all multicast routing protocols construct a tree based topology. So one packet loss on an upstream node means many packet losses on all downstream nodes [8].

In order to achieve reliable broadcast communication, various Media Access Control (MAC) multicast protocols have been recently proposed [9], [10]. Some of these schemes extend basic IEEE 802.11 control mechanisms, such as RTS/CTS and ACK for unicast communication, to broadcast communication. However, these schemes require modifications of a frame format or a hardware. Additionally, routing mechanisms for solving these issues have not been considered in detail [11].

In this paper, we propose an autonomous transmission timing control scheme for collision avoidance in ad hoc multicasting. In the proposed scheme, each node constructs a neighbor-node list to collect neighbor node information. Then, it informs the neighbor-node list to its own upstream node. The upstream node detects its hidden nodes by checking the neighbor-node lists from its downlink nodes. Then, it controls its transmission timing not to conflict hidden node's transmission timing. As a result, the proposed scheme can avoid packet corruption due to hidden node problems. The numerical results show that the proposed scheme can achieve the high delivery ratio.

2 ODMRP

In the proposed scheme, we utilize an On-Demand Multicast Routing Protocol (ODMRP) [12] as a base multicast routing protocol. ODMRP is a mesh based routing protocol for ad hoc multicasting, and uses the forwarding group concept. ODMRP builds routes on demand and uses a mesh to create multicast routes. A soft-state approach is taken in ODMRP to maintain multicast members.

When a multicast source has packets to send, it broadcasts a Join-Query control packet to a entire network. Join-Query packets are periodically broadcast to refresh membership information and update routes. When intermediate nodes receive Join-Query packets, they store a source node ID and a sequence number in its message cache to detect any duplication of Join-Query packets. The routing table is updated with an upstream node ID from messages which were received for a reverse path back to the source node. If Join-Query packets are not a duplicate and a Time-To-Live (TTL) value is greater than zero, they will be rebroadcast.

When multicast destinations receive Join-Query packets, they create and transmit a Join-Reply control packet to their upstream node. When nodes receive the Join-Reply packets, they check whether their own node ID matches the next hop node ID within the Join-Reply packets. If it does, the nodes recognize that they should be forwarding group nodes. Therefore, they set a FG-flag and broadcast the Join-Reply packets. The Join-Reply packets are propagated by each forwarding group member until they reach the multicast source node.

3 Transmission Timing Control Scheme

In ad hoc multicasting, some forwarding group nodes forward same data packets simultaneously. Therefore, transmitted data packets will be corrupted if hidden node problems occur between forwarding group nodes.

Figure 1 shows the example communication with hidden node problems. In this figure, the source node communicates with two intermediate nodes. The first intermediate node communicates with the first destination node and the second destination node. The second intermediate node communicates with the second destination node and the third destination node. In this example, hidden node problems occur between the first intermediate node and the second intermediate node.

In ad hoc multicasting, intermediate nodes forward data packets from upstream nodes. Therefore, the first intermediate node and the second intermediate node forward the same data packets from the source node simultaneously in Fig. 1. As a result, the first destination node can receive the data packet from the first intermediate node, and the third destination node can receive it from the second intermediate node. But, the second destination node cannot receive it due to the packet collision according to hidden node problems.

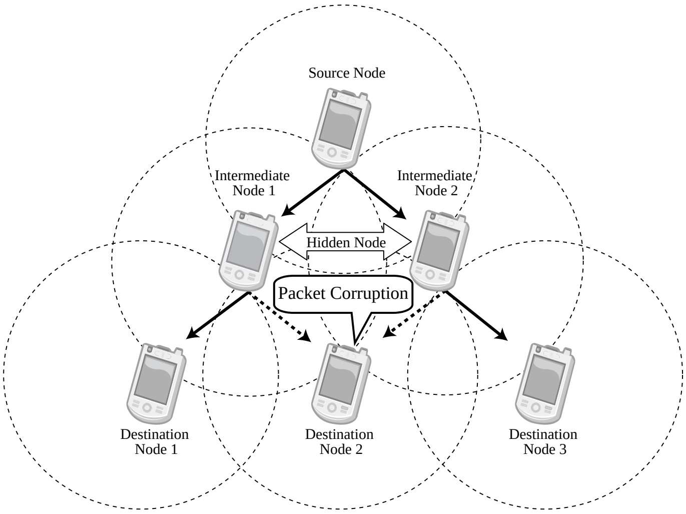

Figure 1: Data packet corruption due to hidden node problems.

In this paper, we propose the autonomous transmission timing control scheme for collision avoidance in ad hoc multicasting. The proposed protocol is extended based on ODMRP to detect hidden nodes, and controls transmission timing not to conflict data packet transmission.

3.1 Construction of neighbor-node list

In order to detect hidden nodes at intermediate nodes, all downstream nodes construct a neighbor-node list in the proposed protocol. The neighbor-node list is constructed due to received Join-Query packets. Figure 2 shows a flow chart when a node receives Join-Query packets. If the node receives Join-Query packets, it stores a previous hop IP address into its own neighbor-node list. Then, it tries to detect any duplication by checking a sequence number within the Join-Query packets. If the received Join-Query packet is not duplicate, the node stores a hop count value into a message cache. Finally, it increments the hop count value in the Join-Query packet. Then, the Join-Query packet will be rebroadcast. If the node is destination nodes, it continues the reception of Join-Query packets for a given length of time. As a result, it collects neighbor node information and constructs the neighbor-node list. After a certain period of time, it transmits a Join-Reply packet including the neighbor-node list to its own upstream node.

3.2 Notification of the neighbor-node list

If a destination node receives Join-Query packets, it broadcasts a Join-Reply packet including a neighbor-node list and

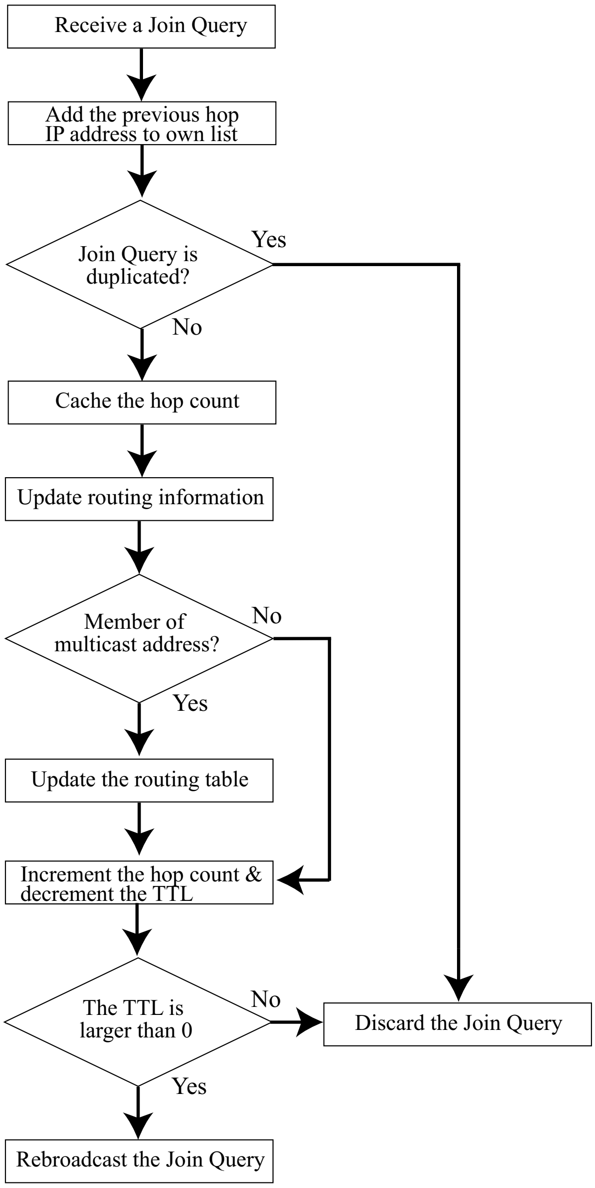

Figure 2: Flow chart for Join-Query packet.

its own hop count value. Therefore, we extend a frame format of the Join-Reply packets in Fig. 4. The new frame format of the Join-Reply packets has fields for neighbor IP addresses. Figure 3 shows a flow chart when a node receives Join-Reply packets. If a node receives Join-Reply packets, it compares its own hop count value with the hop count value within the Join-Reply packets. If the hop count value in Join-Reply packets is larger than its own hop count value, the node recognizes that the Join-Reply packets were broadcasted by downstream nodes. In this instance, it then compares the own neighbor-node list and the neighbor-node list within the Join-Reply packets. If it detects new node addresses, which are not included in its own neighbor-node list, the detected new nodes are hidden nodes against own node. Therefore, data packets may be corrupted at downstream nodes. In the proposed protocol, each node selects delay duration according to an index value of its own node address in the neighbor-node list. The delay duration is calculated by the following equation.

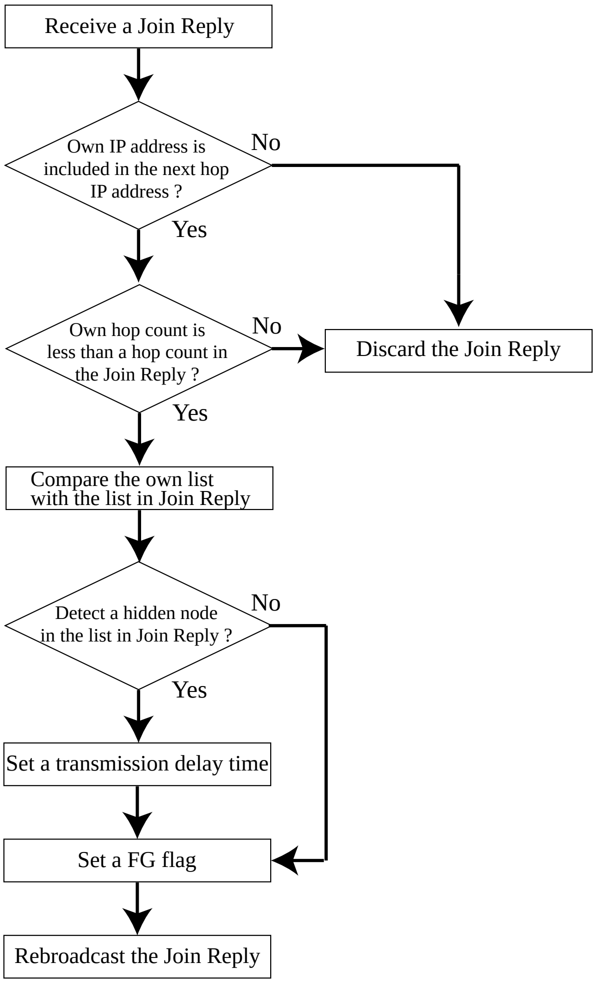

Figure 3: Flow chart for Join-Reply packet.

$$T_{wait} = T_{Trans} \times N_{index} \times \alpha, \tag{1}$$

where T_{Trans} is the transmission duration of data packets, N_{index} is the index number of its own node address in the neighbor-node list of the Join-Reply packets, and α is the constant value for delay-duration margin not to overlap transmission duration of each node.

In addition, hidden nodes also detect their own node address in the neighbor-node list. Therefore, they also set the delay duration according to the index value of the hidden node address in the neighbor-node list. Furthermore, nodes select a new index value not to conflict transmission timing in different neighbor-node lists when it receives some Join-Reply packets from some hidden nodes. As a result, nodes with hidden node problems can select the different delay-duration according to neighbor-node lists from downstream nodes.

Moreover, the source node broadcasts Join-Query packets

<table>
<tr><td>Type</td><td>Count</td><td>R</td><td>F</td><td>Hop Count</td></tr>
<tr><td colspan="5">Multicast Group IP Address</td></tr>
<tr><td colspan="5">Previous Hop IP Address</td></tr>
<tr><td colspan="5">Sequence Number</td></tr>
<tr><td colspan="5">Neighbor IP Address[1]</td></tr>
<tr><td colspan="5">Sender IP Address[1]</td></tr>
<tr><td colspan="5">Next Hop IP Address[1]</td></tr>
<tr><td colspan="5">Route Expiration Time[1]</td></tr>
<tr><td colspan="5">⋮</td></tr>
<tr><td colspan="5">Neighbor IP Address[n]</td></tr>
<tr><td colspan="5">Sender IP Address[n]</td></tr>
<tr><td colspan="5">Next Hop IP Address[n]</td></tr>
<tr><td colspan="5">Route Expiration Time[n]</td></tr>
</table>

Figure 4: Frame format of Join-Reply packet.

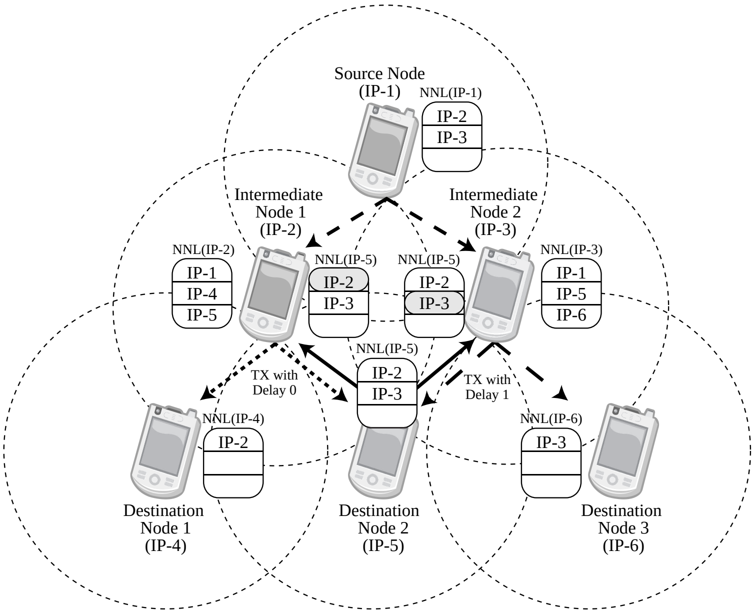

Figure 5: Example operations.

periodically to update routing information. The delay duration is also changed if new Join-Reply packets, which are related to new Join-Query packets, are received. Therefore, the proposed protocol can update appropriate delay duration if node topology is changed by node movement.

3.3 Data delivery

In the proposed protocol, we focus on transmission timing of intermediate nodes with same hop count value from a source node. This is because a hop count value is same, reception timing of data packets is also same. Therefore, intermediate nodes transmit same data packets at same instance if relation between intermediate nodes is hidden node condition. Consequently, intermediate nodes delay data packet forwarding according to set up delay duration. The delay duration is set up by exchanging Join-Query packets and Join-Reply packets. As a result, nodes can avoid data packet collisions at downstream nodes if they have hidden nodes. Additionally, delay time in the proposed protocol is not long because the delay duration is set as multiple transmission duration for one packet and margin duration.

3.4 Example operations

Figure 5 shows example operations of the proposed protocol. In this figure, dot circles mean the communication ranges of each node, arrowed dot lines mean the Join-Query packets, and arrowed lines means the Join-Reply packets.

In this example, the source node can communicate with the first intermediate node and the second intermediate node. But, these two intermediate nodes cannot communicate with each other directly. Therefore, hidden node problems occur between the first intermediate node and the second intermediate node. The first intermediate node can communicate with the first destination node and the second destination node. The second intermediate node can communicate with the second destination node and the third destination node.

The two intermediate nodes transmit the data packets if the source node transmits the data packet in multicasting. The first destination node and the third destination node can receive the data packet. However, the second destination node cannot receive the data packet due to collision if the first intermediate node and the second intermediate node transmit the data packets at the same instant.

In the proposed protocol, the source node broadcasts the Join-Query packet. Then, the first intermediate node and the second intermediate node rebroadcast the Join-Query packets. Finally, all destination nodes also rebroadcast them. As a result, each node can construct the neighbor-node list like as the lists in Fig. 5.

In this example, we assume that the first intermediate node broadcasts the Join-Query packet first, and the second intermediate node broadcasts it secondly. This is because transmission timing of each node is selected due to carrier sense multiple access (CSMA) mechanisms. In the assumption, the second destination node selects the first intermediate node as an upstream node. If the first intermediate node receives the Join-Reply packet from the second destination node, it compares hop count values of its own node and the second destination node. In this figure, the hop count value of the first intermediate node is one and the hop count value of the second destination node is two. Therefore, the first intermediate

Table 1: Simulation parameters.

Simulator	QualNet
Simulation time	300 [s]
Simulation trial	100 [times]
Number of nodes	100 [nodes]
Node positions	Random
Node mobility	None
Area	1000×1000, 1250×1250 [m]
Application	CBR 128K [bps]
Size of data packet	1024 [Byte]
Transmission interval	65 [ms]
Queuing delay	15 [ms]
Refresh time	5 [s]
Wireless device	IEEE 802.11g
Transmission rate	54M [bps]
Transmission range	200 [m]
Propagation model	Free space
Wireless environment	AWGN
Routing protocol	ODMRP, Proposed protocol

node compares the neighbor-node list of its own node and the neighbor-node list in the Join-Reply packet from the second destination node.

In the example, the neighbor-node list of the first intermediate node includes IP-1, IP-4 and IP-5, and the neighbor-node list of the second destination node includes IP-2 and IP-3. As a result, the first intermediate node can find that the neighbor-node list of its own node does not include the node address of IP-3, and it recognizes the second intermediate node with the node address of IP-3 according to the Join-Reply packet from the second destination node. The index number of the address IP-2 in the neighbor-node list is zero. So, the delay duration of the first intermediate node is set to 0 [s].

4 Numerical results

In this section, we compare performance for the proposed protocol with that for the conventional ODMRP protocol. The simulations are performed by the network simulator QualNet [13]. In the simulations, we assume the IEEE 802.11g as the wireless communication device, and the transmission rate is fixed at 54M [bps]. One hundred nodes are placed randomly in 1000×1000 [m] or 1250×1250 [m] area. The source and the destination nodes are selected randomly. The application is a constant bit rate (CBR) with 128K [bps] and data packets with the length of 1024 [Byte] are transferred for 300 [s]. We consider additive white gaussian noise (AWGN) environment and a free space propagation model. In the proposed protocol, nodes receive some Join-Query packets and stores them for queuing delay to construct a neighbor-node list. The queuing delay is set to 15 [ms] in the simulations. The simulation results are an average of 100 trials of simulation. Simulation parameters are shown in detail in Table 1.

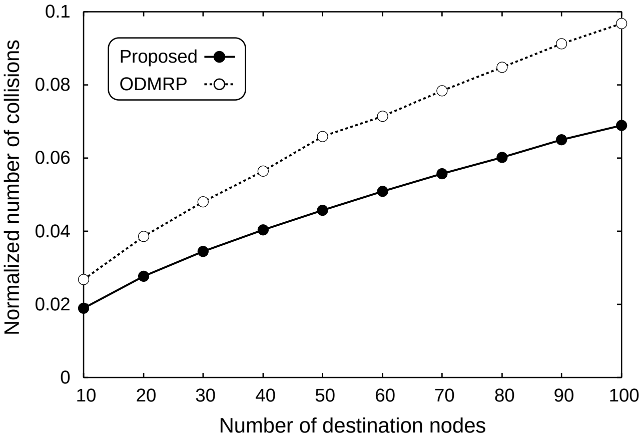

Figure 6: Normalized collisions at multicast group(1000×1000[m])

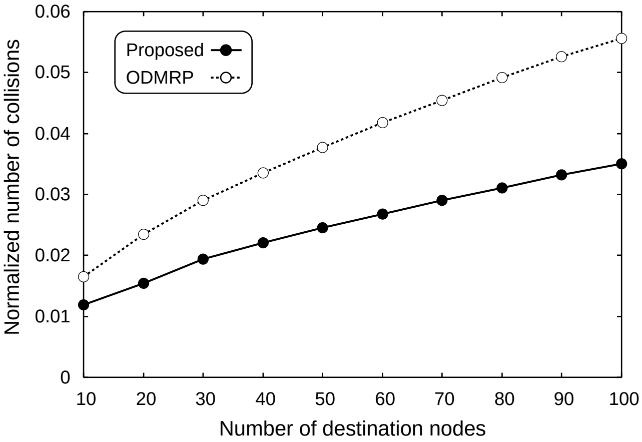

Figure 7: Normalized collisions at multicast group(1250×1250[m])

Figures 6 and 7 show the normalized number of collisions at the multicast group nodes. The multicast group nodes mean the group of the source node, intermediate nodes, and destination nodes. The normalized number of collisions defines that the number of collision packets from the multicast group nodes is divided by the number of transmitted packets from the multicast group nodes. From results, we can find that the number of collisions increases according to increasing in the number of destination nodes. The reason for this is that the number of multicast group nodes also increases, and the number of transmitted packets also increases. Therefore, the packet collision probability also increases. From these figures, the proposed protocol can reduce the number of colli-

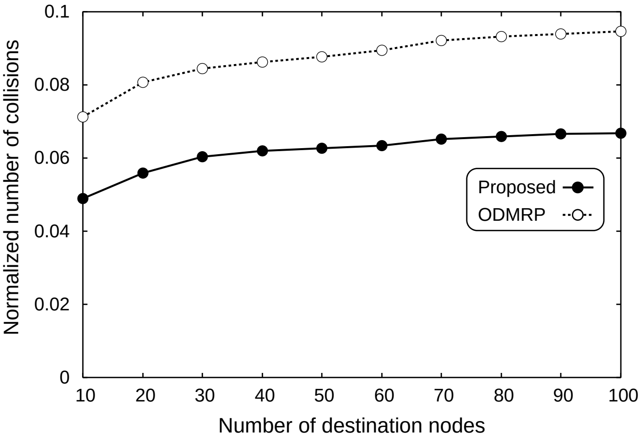

Figure 8: Normalized collisions at all nodes(1000×1000[m])

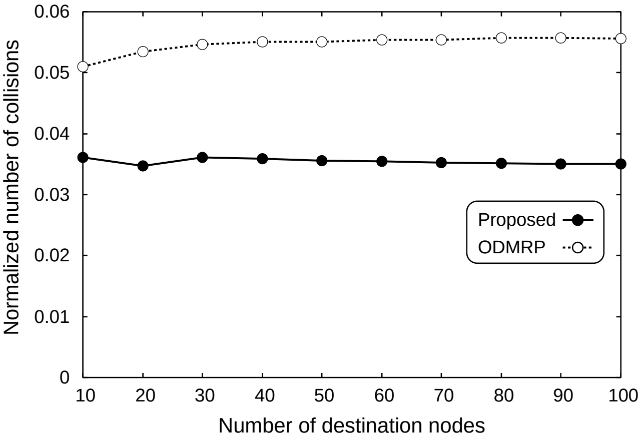

Figure 9: Normalized collisions at all nodes(1250×1250[m])

sions. Moreover, the performance is improved more according to increasing in the number of destination nodes. This is because the proposed protocol can detect hidden nodes and can control the transmission timing not to conflict data packets. Additionally, the number of hidden nodes also increases according to increasing in the number of destination nodes.

Figures 8 and 9 show the normalized number of collisions at all nodes. The normalized number of collisions defines that the number of collision packets from all nodes is divided by the number of transmitted packets from all nodes. From results, it is shown that the proposed protocol can reduce number of collision packets. Moreover, the number of collision packets increases a little according to increasing in the number of destination nodes. This is caused by the proposed protocol can recognize hidden nodes better according to increasing of the destination nodes.

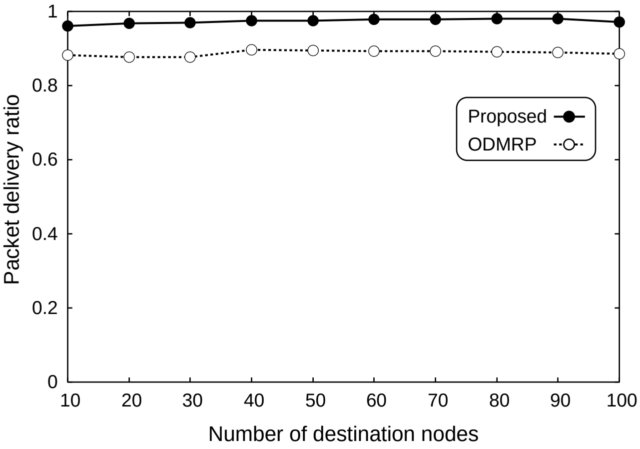

Figure 10: Packet delivery ratio(1000×1000[m])

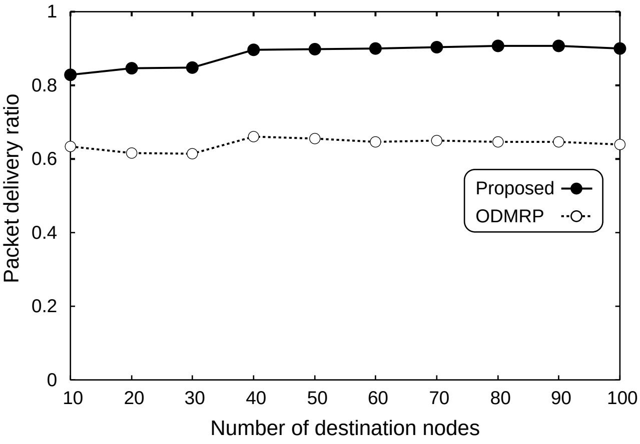

Figure 11: Packet delivery ratio(1250×1250[m])

Figures 10 and 11 show the packet delivery ratio. From results, the proposed protocol can improve the packet delivery ratio more than ten percent. In ad hoc multicasting, a packet corruption at the upstream node means the packet losses at the downstream nodes. Therefore, it is important to avoid the packet corruption at upstream nodes.

5 Conclusions

In this paper, we have proposed an autonomous transmission timing control scheme for collision avoidance in ad hoc multicasting. The proposed protocol is extended protocol based on ODMRP. The node can recognize hidden nodes for its own node by exchanging the neighbor-node list between nodes. Then, it can autonomously control the adequate transmission timing to avoid collisions between hidden nodes and own node. From simulation results, it is shown that the proposed protocol can improve the packet delivery ratio by reducing the number of collisions in the whole networks.

Acknowledgment

This work was supported by Grant-in-Aid for Young Scientists (B)(20700059), Japan Society for the Promotion of Science (JSPS).

REFERENCES

[1] H. Y. Hsieh and R. Sivakumar, "Performance comparison of celluar and multi-hop wireless networks: A quantitative study," Proc. ACM SIGMETRICS, pp. 113–122, Boston, MA, USA, June. 2001.

[2] C. d. M. Cordeiro, H. Gossain, and D. P. Agrawal, "Multicast over Wireless Mobile ad hoc networks: Present and Future Direction," IEEE Network January/February 2003.

[3] K. Viswanath, K. Obraczka, and G. Tsudik, "Exploring Mesh and Tree- based Multicast. Routing Protocols for MANETs," IEEE Transactions on Mobile Computing, vol. 5, no. 1, pp. 28–42, Jan. 2006.

[4] K. Tang, and M. Gerla, "Random Access MAC for Efficient Broadcast Support in ad hoc networks," IEEE Wireless Communications and Networking Conference (WCNC 2000), pp. 454–459, Sept. 2000.

[5] S. Gupta, V. Shankar, and S. Lalwani, "Reliable Multicast MAC Protocol for Wireless LANs," IEEE ICC 03, vol. 1, pp. 93–97, 2007.

[6] W. Si, and C. Li, "RMAC: A Reliable Multicast MAC Protocol for Wireless adhoc networks," IEEE International Conference on Parallel Processing (ICPP 04), vol. 1, pp. 494–501, Aug. 2004.

[7] C.Y. Chiu, E.H.-K. Wu, Y. Wang, and G.-H. Chen, "A Reliable and Efficient MAC Layer Broadcast Protocol for Mobile ad hoc networks," IEEE Transactions on Vehicular Technology, vol.56, no.4, pp.2296–2305, July 2007.

[8] X. Wang, L. Wang, Y. Wang, and D. Gu, "Reliable Multicast Mecha- nism in WLAN with Extended Implicit MAC Acknowledgment," IEEE Vehicular Technology Conference 2008, pp. 2695–2699, 2008.

[9] K. Lee and D. Cho, "A multiple access collision avoidance protocol for multicast service in mobile ad hoc networks," IEEE Communications Letters, vol. 7, no. 10, pp. 508–510, Oct. 2003.

[10] K. S. Lau and C. W. Pao, "Collision Avoidance and Recovery for Multicast Communications in ad hoc networks," IEICE transactions on communications, vol. E88-B, no.7 pp. 2937–2943, Jul. 2005.

[11] M. Pandey and D. Zappala, "A Scenario-Based Evaluation of Mobile Ad Hoc Multicast Routing Protocols," July. 2004.

[12] Sung-Ju Lee, William Su, Mario Gerla, "On-Demand Multicast Routing Protocol (ODMRP) for ad hoc networks," Internet-Draft, draft-ietf-manet-odmrp-02.txt, Jan. 2000.

[13] Network Simulator Qualnet is http://www.scalable-networks.com/