Abstract—This paper assesses the application–level QoS of live audio and video multicasting in a wireless ad hoc network. We regard the quality of media synchronization as the major part of the application–level QoS. We also investigate the inter–destination synchronization quality of three schemes: the master–slave destination scheme, the synchronization maestro scheme, and the distributed control scheme. The inter–destination synchronization adjusts the output timing among destinations in a multicast group. We perform computer simulation with a string topology network. In the simulation, a source terminal multicasts the media streams to six destinations by using ODMRP (On–Demand Multicast Routing Protocol), which is a multicast routing protocol for ad hoc networks. The simulation result shows that the master–slave destination scheme, which sends smaller numbers of control packets for inter–destination synchronization control than the other two schemes, provides higher quality of application–level QoS; however, the QoS of this scheme is sensitive to the location of the master destination. When it is difficult to find an appropriate destination for the master, the synchronization maestro scheme can be the second best choice.

I. INTRODUCTION

Wireless ad hoc networks have gained considerable interest in recent years. They are networks with no fixed infrastructures, such as underground cabling or base stations, where all nodes are capable of moving and can be connected dynamically in an arbitrary manner. Each mobile host acts as a router, which discovers and maintains routes to other hosts and forwards packets for them in the network.

Some applications of ad hoc networks require the ability to support real–time multimedia streams such as live audio and video over the network. Examples of such applications include audio–video streaming, multimedia conference, video chat, and remote monitoring. These applications are often performed in one–to–many or many–to–many communications. Hence, multicasting is one of the most important techniques for these applications.

In ad hoc networks, the routing protocol is very important to multicasting of media streams. This is because the network topology dynamically changes owing to the terminal movement, and then the quality of transmitted media streams drastically changes.

A number of ad hoc routing protocols for multicasting have been proposed over the past few years [1]. Most of the previous studies on the ad hoc multicast routing have focused on the efficiency of multicast communications such as packet delivery ratio and protocol overhead [2]. However, these studies do not have a well defined framework to provide QoS support for multimedia applications.

When we multicast continuous media streams such as audio and video in ad hoc networks, the temporal structure of the streams may be disturbed largely by delay and its jitter. In wireless networks such as IEEE 802.11, terminals share the same physical channel. Furthermore, wireless networks have slower transmission rates than wired ones. Thus, its delay easily increases, and then its throughput largely decreases. In addition, as in IEEE 802.11, the media access control (MAC) protocol usually has a carrier–sensing capability and a retransmission–based error recovery mechanism in order to recover transmission errors in the wireless channel. This also increases network delay and its jitter. Thus, in order to preserve the temporal relation, we need the media synchronization control [3], which is one of the application–level QoS control.

We identify three types of media synchronization: intra–stream synchronization, inter–stream synchronization and inter–destination (or group) synchronization. The intra–stream synchronization control is necessary for the preservation of the timing relation between media units (MUs) such as video frames in a single media stream; an MU is the information unit for media synchronization. The inter–stream synchronization is required for keeping the temporal relations among MUs in multiple media streams. The inter–destination synchronization adjusts the output timing of each MU multicast to two or more destinations so that the MU can be output simultaneously at all the destinations. It is an indispensable function to support some types of multicast applications; also, it is necessary to realize the fairness among destinations in many multicast applications.

Live audio and video streaming can be one of the most promising applications even in ad hoc networks, and then we should assess its quality. For the users, the subjective quality (i.e., user–level QoS) is the most important QoS; it is closely related to application–level QoS. The preservation of the temporal structure is essential to high quality of the application–level QoS of continuous media [4]. However, in the literature, we cannot find any study on the continuous media multicasting in ad hoc networks from an application–level QoS point of view.

For the QoS support in ad hoc networks, a few studies modify the MAC protocol [5], [6]. In those studies, continuous media are transmitted with higher priority than discrete media like computer data. They show the effectiveness of the proposed protocols by the network–level or end–to–end–level QoS assessment in terms of packet throughput, delay and loss rate. However, the application–level QoS assessment is not performed. In addition, they do not mention multicast applications.

On the other hand, in [7], Ruiz et al. propose an application–level QoS control scheme for multicast multimedia streaming. In order to preserve good quality of user–level QoS, it changes codec, code–specific parameters and frame sizes of audio and video according to the network condition, which is measured by the percentage of packet loss. They apply the control to audio and video multicasting in ad hoc network. They show the effectiveness by the loss rate and the maximum delay jitter. However, they do not assess the temporal quality of output continuous media. Furthermore, they treat a one–to–one (i.e., unicast) video conference in the performance evaluation, and then the inter–destination synchronization problem is not taken into account.
The application-level QoS of continuous media multicasting has been assessed in other kinds of networks [8], [9]. In [8], the authors compare the application-level QoS of three inter-destination synchronization schemes in a wired network: the master–slave destination scheme, the synchronization maestro scheme, and the distributed control scheme. These schemes are based on the virtual-time rendering (VTR) media synchronization algorithm [10], which is applicable to networks with unknown delay bounds by dynamically adjusting the MU buffering time according to the network condition. Furthermore, in [9], Tasaka et al. examine the influence of handover on the application-level QoS including inter-destination synchronization quality in an integrated wired and wireless network. However, the characteristics of these schemes in ad hoc networks have not been clarified.

The purpose of this paper is the assessment of the application-level QoS of the three inter-destination synchronization schemes in ad hoc networks. We develop a simulation model with a static string network topology for simplicity of the assessment. That is, the topology does not have any redundant routes, and the nodes do not move. However, this simplistic assumption allows us to get a better understanding of the basic characteristics of these schemes in the ad hoc network.

The rest of this paper is organized as follows. Section II explains routing protocols in ad hoc networks. Section III describes principles of the inter-destination synchronization schemes. Section IV illustrates a methodology for the application-level QoS assessment, including the network configuration, simulation method and QoS parameters. The simulation results are presented and discussed in Section V.

II. MULTICAST ROUTING PROTOCOL

Well established routing protocols offer efficient multicasting service in conventional wired networks. For example, PIM (Protocol Independent Multicast) [11] is one of the most famous protocols. However, these protocols may fail to keep up with node movements and frequent topology changes in an ad hoc network. An efficient multicast over an ad hoc network requires dynamic group membership and constant updates of delivery path due to node movements.

One straightforward way to provide multicast in an ad hoc network is flooding [1]. It is suggested that in a highly mobile ad hoc network, flooding in the whole network may be a viable alternative for reliable multicast. However, this approach has considerable overhead. A number of duplicated packets are sent, and then packet collisions occur in a multiple-access-based ad hoc network. In addition, within a wireless medium, it is even more crucial to reduce transmission overhead and power consumption. Thus, flooding is not appropriate for many ad hoc networks.

In this paper, we employ ODMRP (On-Demand Multicast Routing Protocol) [12] as the multicast routing protocol. This protocol is one of the most efficient multicast routing protocols in ad hoc networks [2] and is now found in the Internet Draft. ODMRP is a mesh-based protocol that uses a forwarding group concept; that is, only a subset of nodes forwards the multicast packets. It applies on-demand procedures to build routes dynamically and maintain multicast group membership.

If a multicast source has packets to send when it does not have any route to the multicast group, it broadcasts a Join—Query control packet with piggybacked data payload to the entire network. When a node receives a non-duplicate Join—Query, it stores the upstream node ID (i.e., backward learning) and rebroadcasts the packet.

When a Join—Query reaches a multicast receiver, it creates and broadcasts a Join—Reply control packet to its neighbors. The Join—Reply packet has a table which includes pairs of a source node ID and a next hop node ID. When a node receives a Join—Reply, it checks if there exists some entry where the node’s ID matches the next hop node ID. If it exists, the node realizes that it is on the path to the source and thus is a part of the forwarding group. It then broadcasts its own Join—Reply built on matched entries. Each forwarding group member propagates the Join—Reply until it reaches the multicast source via the shortest path. This process constructs the routes from sources to receivers and builds a mesh of nodes; it is the forwarding group.

After this group establishment and route construction process, a source can multicast packets to receivers via selected routes and forwarding groups. In ODMRP, the source multicasts subsequent packets by using broadcast frames of the underlying MAC protocol.

In ODMRP, no explicit control packets need to be sent to join or leave the group; multicast sends refresh the membership information and update the routes by sending a Join—Query packet periodically.

III. INTER-DESTINATION SYNCHRONIZATION SCHEMES

In this paper, we employ the three inter-destination synchronization schemes: the master—slave destination scheme, the synchronization maestro scheme, and the distributed control scheme. These are based on the VTR media synchronization algorithm.

The VTR algorithm adaptively changes the buffering time according to the network condition. Initially, the buffering time is set to a rough estimate of the maximum delay jitter, which is denoted by $J_{max}$; this value may be different from destination to destination. When inter-destination synchronization control is applied, however, a constant delay value $\delta$ instead of individual buffering times $J_{max}$ is used commonly to all the destinations; this is referred to as the target delay time $[13]$, which is defined as the time from the moment an MU is generated until the instant the MU should be output. After the first MU is received, the buffering time is changed by the modification of the target output time of each received MU. The target output time is the time when an MU should be output. If the network condition differs from destination to destination, the target output time may be different. Thus, we need the inter-destination synchronization control in order to adjust the target output time at all the destinations. In what follows, we outline the basic idea of the three inter-destination synchronization schemes. For details, see [8] and [13].

In the master—slave destination scheme, destinations are grouped into a master destination and slave destinations. Each slave destination does not send any information on the output timing. It adjusts the target output time of MUs to that of the master destination. Only the master destination multicasts its output timing to all the slave destinations.

The synchronization maestro scheme employs a synchronization maestro, which can be the source or one of the destinations. It gathers the information on the output timing from all destinations and adjusts the output timing among the destinations by distributing control packets. In order to do this, each destination unicasts the information to the maestro, and the maestro multicasts the adjusted output timing.

In the distributed control scheme, each destination decides the reference output timing from among the output timing of itself and that of the other destinations. In this scheme, all the destinations multicasts the control packets. Hence, bursty traffic due to the control packets may degrade the output quality of media streams. Thus, each destination sets a random backoff timer before sending a control packet.
IV. METHODOLOGY FOR QUALITY ASSESSMENT

We compare the application–level QoS of the inter–destination synchronization schemes by computer simulation with ns–2 (network simulator version 2) [14]. We use an implementation of ODMRP developed by the Rice University Monarch Project [15].

A. Network Configuration

Figure 1 illustrates the network configuration in the simulation. In this paper, we employ 11 wireless terminals, which construct a string topology. The interval between two adjacent terminals is constant, 15 m. The source terminal (MS) is located in the center of the network. We deploy six destination terminals (MR1 through MR6) and four load terminals (LT1 through LT4) as shown in Fig. 1.

In this paper, we have chosen a quite simple topology as the first step of the study. This simplistic assumption provides a good insight into the basic characteristics of the inter–destination synchronization schemes in ad hoc networks. However, it imposes several limitations on the assessment result. For example, we cannot address the link failure problem caused by mobility. In addition, there is no redundant route in the network. Thus, our assessment does not include the influence of the routing algorithm of ODMRP. Hence, we plan to extend our assessment so as to incorporate more realistic situations that are representative of the real world, such as more pragmatic topologies, mobile nodes and varying node distances.

We consider a detailed simulation model which is based on the distributed coordination function (DCF) [16] of the IEEE 802.11 wireless LAN. We employ the free space propagation model implemented in ns–2. Each terminal has an omni–directional antenna. The radio model uses characteristics similar to a commercial radio interface, Lucent Technologies’ Orinoco 802.11b 11 Mbps PC card, that is, we assume IEEE 802.11b. In the simulation, the transmission range of each terminal is about 22.49 m as shown by circles in Fig. 1. That is, nodes are only within range of their immediate neighbors.

B. Method of Simulation

We assume MS as the voice and video sources. MS multicasts the media streams to MR1 through MR6 with RTP/UDP. We use a voice stream of ITU–T G.711 μ–law and an MPEG1 video stream. Table I shows the specifications of the voice and video.

We take the media capturing and encoding delay time into consideration in the simulation. The capture duration of a video MU is just a moment. However, it spends much time to encode a video frame. In this paper, we set the capturing and encoding delay time of each video MU to 8 ms, which is the same time as that in the experimental system in [17]. Each MU leaves the source the capturing and encoding delay time after its timestamp.

In the simulation, we assume the globally synchronized clock [10] and compare the application–level QoS of five schemes: NC (No Control), VTR–TDT, Maestro, Master–Slave, and Distributed. NC means that no media synchronization control is carried out. VTR–TDT implies the VTR algorithm with the knowledge of the target delay time. This scheme does not employ any inter–destination synchronization mechanism. However, all the destinations with this scheme have the same target delay time δ; that is, there is a consensus among them about the target delay time. Therefore, each destination sets the initial buffering time of VTR individually so as to have the same target delay time. Maestro and Master–Slave denote the synchronization maestro scheme and the master–slave destination scheme, respectively. Distributed means the distributed control scheme with the random backoff timer.

In [8], we notice that in a wired network, the synchronization maestro should be deployed in order to prevent the competition between the control packets from each destination to the synchronization maestro and MUs; this is satisfied by selecting the media source as the maestro. We have also found that the most heavily loaded destination should be selected as the master destination. On the basis of the results, we choose the source terminal MS as the synchronization maestro in Maestro. We also select the destination terminal MR6, which is the most heavily loaded destination in the simulation, as the master in Master–Slave.

In the simulation, we set the target delay time δ to either 50 or 100 ms. The other thresholds and parameters in the VTR algorithm and the inter–destination synchronization control have the same values as those in [8].

LT1 through LT4 are used to handle traffic flows of interference. We also employ ODMRP for routing of the load traffic. LT1 sends/receives the traffic to/from LT2. LT3 and LT4 also handle the traffic in the same way as LT1 and LT2. The load terminals generate fixed–size IP datagrams of 1500 bytes each at exponentially distributed intervals. The amount of the interference traffic is adjusted by changing the average of the interval. We set the same average of the interval for all the load terminals. Then, we refer to the average amount of the interference traffic for each load terminal as the average load.

C. QoS Parameters

In order to assess the application–level QoS of the inter–destination synchronization schemes, we need to examine the inter–destination synchronization quality as well as the intra–stream and inter–stream synchronization quality.

In [17], JPEG is employed for video codec. On the other hand, this paper handles MPEG video. However, because of the GOP pattern in this paper, we have assumed that the capturing and encoding delay time of each MU is approximately the same as that of JPEG video in [17].
For the inter–destination synchronization quality, we evaluate the mean square error of inter–destination synchronization\(^{2}\). In two destinations \(A\) and \(B\), it denotes the mean square of the difference between the output time of an MU (excluding skipped MUs) at destination \(A\) and that of the MU at destination \(B\). In this paper, we suppose six destinations. Thus, there are many combinations of two destinations. However, some combinations have the same tendency as other combinations or very small inter–destination synchronization error. Therefore, in this paper, we select a reference destination from among all the destinations and then calculate the average of mean square errors of inter–destination synchronization between the reference destination and another one; we use it for quality assessment.

On the other hand, for the quality assessment of intra–stream synchronization for audio or video, we evaluate the coefficient of variation of output interval, which represents the smoothness of output of a media stream. The mean square error of intra–stream synchronization also shows the intra–stream synchronization quality; it denotes the average square of the difference between the output time difference of two consecutive MUs and their timestamp difference. In addition, we use the MU loss rate, which is the ratio of the number of MUs lost to the total number of MUs generated.

For the inter–stream synchronization quality, we calculate the mean square error of inter–stream synchronization, which is defined as the average square of the difference between the output time of each slave MU and its derived output time. The derived output time of each slave MU is defined as the output time of the corresponding master MU plus the difference between the timestamps of the two MUs [10].

The average MU delay, which is the average time of MU delay, is a key measure for live media. The MU delay is defined as the time interval from the moment an MU is generated until the instant the MU is output.

V. SIMULATION RESULTS

We first show the inter–destination synchronization quality. Then, we present the intra–stream and inter–stream synchronization quality at MR6. We have also assessed the synchronization quality at MR4 and that at MR5. As a result, we saw that the relationships between the schemes at these terminals are approximately the same as that at MR6.

Each symbol in the figures to be shown represents the average of 10 measured values which were obtained by changing the random seed for generating the interference traffic. We also show 95% confidence intervals of the QoS parameters in the figures. However, when the interval is smaller than the size of the corresponding symbol representing the simulation result, we do not show it in the figures.

A. Inter–Destination Synchronization Quality

Figures 2 and 3 show the average of mean square errors of inter–destination synchronization between MR1 and another destination for voice and that for video, respectively, when \(\delta\) is set to 100 ms. Figure 4 displays the average MU delay of voice at MR6 for \(\delta = 100\) ms. Meanwhile, the average of mean square errors of inter–destination synchronization between MR1 and another destination for video when \(\delta\) is set to 50 ms is shown in Fig. 5, and Fig. 6 depicts that for video.

In Figs. 2 and 3, we find that for all the data loads here, the average of mean square errors of inter–destination synchronization with VTR–TDT is approximately the same as that with Maestro or Master–Slave and is smaller than that with Distributed. This is because \(\delta = 100\) ms is an ideal setting for VTR–TDT, in which all the destinations individually change the buffering time according to the network condition. Then, the buffering time in heavily loaded destinations may increase, while that in lightly loaded destinations may keep the initial buffering time, which is decided by \(\delta\). This paper employs a static string topology, in which the network delay jitter is usually much smaller than that in the network with the node movement. Thus, when we set \(\delta\) to a sufficiently large value so as to absorb the jitter, the buffering time in VTR–TDT hardly increases from the initial buffering time. Furthermore, in VTR–TDT, all the destinations have the same value of \(\delta\). If the value of \(\delta\) is large enough to absorb the difference in the network delay among the destinations, the quality of inter–destination synchronization is kept very high, though too large a value of \(\delta\) destroys the real–time property. In Fig. 4, we notice that for all the data loads here, the average MU delay with VTR–TDT does not increase so largely from \(\delta = 100\) ms. Meanwhile, we notice in Figs. 5 and 6 that when we set \(\delta\) to 50 ms, VTR–TDT has the largest inter–destination synchronization error among all the schemes except for NC. This is because a value of 50 ms for \(\delta\) is insufficient for VTR–TDT. In the simulation, we set the capturing and encoding delay time of a voice MU to 50 ms. In other words, the value is the lower limit of the MU delay of voice, and then almost all the destinations cannot absorb the jitter with the initial buffering time decided

\[^{2}\text{It is not clear how large mean square error of inter–destination synchronization is allowable for applications. In order to answer this question, we need to carry out subjective assessment in a systematic way.}\]
by $\delta = 50$ ms. Thus, in order to absorb the jitter, each destination changes the buffering time individually in VTR–TDT; this degrades the inter–destination synchronization quality. On the other hand, the inter–destination synchronization schemes can control the buffering time at all the destinations in order to adjust the output timing among them by telling the information on the appropriate output timing, even if we select an insufficient value of $\delta$.

From the above observations, we can say that the inter–destination synchronization quality of VTR–TDT is sensitive to the value of $\delta$. In order to achieve high quality inter–destination synchronization of VTR–TDT, we must set $\delta$ to an appropriate value. However, the appropriate value of $\delta$ is usually unknown, and it is very difficult to choose the value, because the network topology dynamically changes owing to the node movement in ad hoc networks. Therefore, VTR–TDT is not suitable for multimedia multicasting in ad hoc networks.

Figures 2, 3, 5 and 6 show that for almost all the data loads here, Distributed has the largest inter–destination synchronization error among the three inter–destination synchronization schemes, while Master–Slave has the smallest error. This is because in Distributed, each destination multicasts the control packets for the inter–destination synchronization control to all the other destinations, and then the network becomes heavily loaded. In that situation, each destination may wait for the transmission of packets for a long time to avoid a collision. Therefore, the difference in the network delay among the destinations is large in Distributed. On the other hand, in Master–Slave, only the master destination multicasts the control packets; that is, Master–Slave transmits smaller numbers of packets than Maestro and Distributed. Thus, in Master–Slave, the difference in the network delay among the destinations is not so large as that in Maestro or Distributed. We can confirm that for the data loads heavier than about 600 kbps in Fig. 4, the average MU delay of voice with Distributed increases largely, although that with Master–Slave hardly increases.

In Figs. 2, 3, 5 and 6, we see that for almost all the data loads here, the averages of mean square errors of inter–destination synchronization with all the schemes are smaller than 1000 ms$^2$; this value seems sufficiently small for many applications. However, in the applications which are severe with inter–destination synchronization quality, the errors of this level may not be acceptable. In addition, if the network size becomes large and if the topology becomes complicated, we will be faced with the degradation of inter–destination synchronization quality in many applications. This is left as a subject in future work.

### B. Intra–Stream and Inter–Stream Synchronization Quality

We now show the intra–stream and inter–stream synchronization quality when we set $\delta$ to 100 ms. From the previous results, we saw that VTR–TDT is not appropriate from an inter–destination synchronization quality point of view. Thus, we do not treat VTR–TDT in this subsection. We compare the quality of the four schemes: NC, Maestro, Master–Slave, and Distributed. We have also assessed the synchronization quality with $\delta = 50$ ms. As a result, we saw that the relationships between the schemes with $\delta = 50$ ms are approximately the same as that with $\delta = 100$ ms. Therefore, we show the simulation results only for $\delta = 100$ ms below.

Figure 7 displays the MU loss rate of voice at MR6 versus the data load, and Fig. 8 plots that of video likewise. The mean square error of intra–stream synchronization for voice at MR6 and that for video are shown in Figs. 9 and 10, respectively. Figure 11 depicts the coefficient of variation of output interval for voice at MR6 as a function of the data load. Figure 12 plots the mean square error of inter–stream synchronization at MR6 versus the data load. From Figs. 7 and 8, we notice that when the average load is heavier than around 500 kbps, Maestro, Master–Slave and Distributed, which exert the inter–destination synchronization control, have larger MU loss rates than NC. The reason is as follows. ODMRP uses broadcast frames in the IEEE 802.11 MAC protocol for data transmission. The MAC protocol has an ACK–based
retransmission mechanism. However, no acknowledgment shall be transmitted by any recipients of the broadcast frame. Thus, the source terminal has no idea about the status of the transmitted broadcast frame and then cannot retransmit the frame. That is, when the collision of frames occurs in the MAC layer, the frames are just dropped. Meanwhile, the inter-destination synchronization schemes transmit control packets for its control. These packets cause much collisions, which occur most remarkably in Distributed as mentioned earlier. Thus, the MU loss rate increases as the number of control packets increases.

In Figs. 9 and 10, we find that the mean square error of intra-stream synchronization with NC is the largest among all the schemes. Thus, we can say that the media synchronization control is effective for preserving the temporal structure of media streams. We see in Fig. 11 that for the data loads between about 500 kbps and around 650 kbps, the coefficient of variation of output interval for voice with Distributed is the largest among all the schemes. Furthermore, the coefficient for voice with Maestro is the second largest. This is due to many collisions in the two schemes, which lead to much MU loss as shown in Figs. 7 and 8. The coefficient for video with Distributed, which is not shown in this paper, is also the largest among all the schemes in the heavily loaded condition.

In Fig. 12, we find that the mean square errors of inter-stream synchronization with all the schemes are much smaller than 6400 ms² (= 80² ms²), which is a threshold of high inter-stream synchronization quality reported by Steinmetz [18]. High quality of inter-stream synchronization even with no control is a characteristic of live media [19].

From the above discussion, we notice that the control packets for inter-destination synchronization control affect the output quality of media streams largely, and then Master-Slave is the best scheme among the three inter-destination synchronization schemes. However, Master-Slave has a problem that the most heavily loaded destination is not always known. Owing to this, we cannot always select the master destination appropriately. The inter-destination synchronization quality of Master-Slave is sensitive to the location of the master, though we have not shown any data about this because of limitations of space. Thus, we should use Maestro if it is not clear which destination is the most heavily loaded.

VI. CONCLUSIONS

In this paper, we assessed the application-level QoS of live audio-video multicasting in a wireless ad hoc network based on IEEE 802.11b. The network has a string topology. We compared the quality of three inter-destination synchronization schemes: the master-slave destination scheme, the synchronization maestro scheme, and the distributed control scheme. As a result, we saw that the control packets for inter-destination synchronization control affect the output quality of the media streams largely. The master-slave destination scheme, which sends smaller numbers of control packets for inter-destination synchronization
control than the other two schemes, provides higher quality of application–level QoS. On the other hand, the application–level QoS of the distributed control scheme, in which all the destinations multicast control packets, deteriorates in heavily loaded condition. This is because the IEEE 802.11 MAC protocol provides no reliability for broadcast frames, which are used by ODMRP for data transmission.

In order to solve the problem, some reliable broadcast schemes based on the IEEE 802.11 MAC protocol have been proposed [20], [21]. In addition, we can employ an application–level retransmission–based error recovery [22]. The implementation of these methods is one of our future studies.

In addition, the master–slave destination scheme has a problem that the most heavily loaded destination is not always known, and we cannot always select the master destination appropriately; the QoS of this scheme is sensitive to its location. Thus, if it is not clear which destination is the most heavily loaded, we should use the synchronization maestro scheme. The master selection algorithm in ad hoc networks is also a future study.

Furthermore, our assessment does not include the influence of the routing algorithm of ODMRP because we have employed a simple string topology. Hence, we should assess the application–level QoS in other network configurations which are representative of the real world, such as more pragmatic topologies, mobile nodes and varying node distances.

ACKNOWLEDGMENT

This work was supported by the Grant–In–Aid for Young Scientist (B) of Japan Society for the Promotion of Science under Grant 15760262 and the Grant–In–Aid for Scientific Research (B) under Grant 14350200.

REFERENCES