

PAPER

Performance Improvement of an Ad Hoc Network System for Wireless Data Service

Takayuki YAMAMOTO[†], *Student Member*, Masashi SUGANO^{††}, Masayuki MURATA^{†††},
Takaaki HATAUCHI^{††††}, *Members*, and Yohei HOSOOKA^{††††}, *Nonmember*

SUMMARY In ad hoc wireless networks, wireless terminals can autonomously construct and can maintain the network. They communicate with some neighbor terminals, exchange network information and determine routes for packets on the multi-hop wireless network. Flexible Radio Network (FRN), one of the ad hoc wireless network systems, adopts a proprietary protocol that provides a multiple routes management and a packet retransmission mechanism against packet transmission errors. This system is a commercial product that has been in use in a recent few years. In this paper, we first evaluate the performance through simulations for data-link protocol and routing protocol of the FRN to clarify its basic properties. Furthermore, we propose some techniques that enhance its performance and solve problems on the protocols. We show how they improve the system performance through simulations and analyses.

key words: *ad hoc wireless network, routing protocol, simulation, analysis*

1. Introduction

Ad hoc wireless networks are self-organized networks built with wireless terminals. They communicate with each other and exchange the network information. They can relay a packet for another terminal to construct a wide area multi-hop wireless network. The ad hoc networks need neither a wired backbone network nor base stations. As a result, network installation, expansion and removal can be performed easily and quickly. Such a wireless infrastructure covers a wide range of applications, e.g., distributed computing systems, disaster recovery networks, and sensor networks. Accordingly, many studies have been dedicated to analyze its characteristics and/or propose new routing methods (see, e.g., [1]–[10]).

Flexible Radio Network (FRN) is one of the commercially available products based on ad hoc wireless network system [11]. A large-scale network with stationary terminals can be installed easily into existing

facilities by the FRN. In addition, the network can be extended only by adding the radio terminal if needed. This system is now utilized, for example, for collecting usage information of entrance gates, for monitoring a sales account of vending machines, and for electric energy control in factories. The FRN adopts a proprietary protocol that can efficiently adapt to terminal failures or a change of network configuration. The routing protocol is table-driven and bases on the distance vector algorithm. The difference between the FRN and existing ad hoc routing protocols like DSDV [12] is that the FRN is capable of maintaining multiple routes to each destination and transmitting packets by a roundabout way if the shortest route is temporally unavailable. There are some other multipath routing protocols. Ad-hoc On-Demand Distance Vector with Backup Routes (AODV-BR) [13] and Split Multipath Routing (SMR) [14] are on-demand multipath routing protocols. With these protocols, a terminal performs the route searches by flooding a route request packet. If we assume a network that its topology will not change much frequently, this flooding route search will cost a lot. The FRN can search and maintain multiple routes to every node in the network without any additive messages except for periodic route table exchanges. We introduced this system and investigated its performance in [15]. However, it is not clear how system parameters affect the performance such as throughput and the packet loss rate. In the current system, these are decided by trial and error. In order to clarify the scope of this system, detailed evaluation is necessary.

Next, we propose some techniques for improving the performance of the FRN. These techniques are based on some problems found through a process of simulated performance evaluation and/or an experience in the real environment.

Some of them make the setting of system parameters more appropriately. One target parameter is a *retransmission interval*. Every terminal has a hop-by-hop packet acknowledgement/retransmission mechanism against packet transmission failures. The retransmission interval is selected randomly, however in the original FRN system, there are few options of selection and retransmitted packets often collide again. We simulated some longer retransmission timers and evaluated their effects for the packet loss rate and the

Manuscript received January 15, 1999.

Manuscript revised January 16, 1999.

[†]The author is with Graduate School of Information Science and Technology, Osaka University, Toyonaka-shi, 560-8531, Japan.

^{††}The author is with Osaka Prefecture College of Nursing, Habikino-shi, 583-8555, Japan.

^{†††}The author is with Cyber Media Center, Osaka University, Toyonaka-shi, 560-8531, Japan.

^{††††}The authors are with Energy and Electrical Systems Company, Fuji Electric Co. Ltd., Hino-shi, 191-8502, Japan.

end-to-end packet transmission time. Another parameter targeted in this paper is a *maximum lifetime* of packets. All data packets have the value of the maximum lifetime in their headers and the terminals use them to erase some long-living packets. In the original FRN, the same value that is sufficiently long for a network is selected for all data packets. However, maximum lifetime has close connections in the necessary hop count to the destination node. We can achieve better performance by setting an adaptive lifetime for each packet.

Another proposal is a technique to decrease the number of packet duplications. We have investigated the FRN system and attended to a problem that packets were sometimes duplicated unnecessarily in a packet relaying process. In the FRN, each wireless terminal checks a packet transmission in data-link layer at every hop. Transmission failure can be detected when a terminal does not receive a corresponding acknowledgement from a neighbor terminal within a pre-specified time, that is, the retransmission timeout described above. The problem lies in that the sender terminal recognizes a transmission error whenever it cannot receive the acknowledgement. If the acknowledgement is lost after successful data transmission, the terminal will retransmit the packet although the first transmitted data is not lost. We call this retransmitted packet a duplicated packet, which leads to the higher traffic load than the actual one. To make the matter worse, more duplicated packets are generated as the traffic load becomes higher because more corresponding ACKs are lost. Thus, the network performance degrades rapidly. We propose a technique to enable terminals to recognize and erase the duplicated packets selectively in Section 5.

The remainder of this paper is organized as follows. We first describe the FRN system in Section 2 and evaluate the relation between its performance and the system parameter setting in Section 3. Then, we propose two approaches to set the system parameters more adaptively and evaluate them through simulations and analyses in Section 4. Next, we examine a packet duplication process in detail and propose a technique to decrease the number of duplicated packets in Section 5 and conclude this paper in Section 6.

2. System Description of FRN

2.1 Network Configuration

In the FRN, every wireless terminal is called a *node*. Some nodes with which a node can communicate directly are called *neighbor nodes*. Every node can determine a route for a packet and can relay the packet to one of the neighbor nodes. In more detail, a *host node* generates and receives data packets, and other nodes are called *relay nodes* that construct a multi-

hop network. Every node maintains network structure information in a *network configuration table* that contains the route information from the node itself to each destination node. Each route information consists of the neighbor node's address on the route and the hop count to the destination. When a node has multiple routes to a destination node, they are sorted in order of the hop count. Every node exchanges a network information packet periodically that contains information of the shortest routes from itself to every destination, and updates its configuration table by the packets from neighbor nodes.

2.2 Data-link Protocol

A radio channel is divided into fixed-length time slots. In a wireless network, every neighbor node of a certain node can receive packets from the node even when it is not the source/destination of the packet. The FRN utilizes this property for the hop-by-hop transmission acknowledgment to enhance the network reliability, i.e., the FRN uses a broadcast-based method for packet transmission. See Figure 1 as an example. Figure 1(a) shows a case where packet transmission and acknowledgment at node A succeed. Node B receives the packet from node A and relays it to another node successfully. At the same slot, node A receives this relayed packet although the next hop of the packet is not destined for node A, because node A is within the range of radio transmission from node B. This acknowledgment is called a *relay echo* (or simply an *echo*). If the relay echo is successfully received by node A, node A can delete the packet that stays in its buffer for retransmission. The case where node A fails the first transmission is shown in Figure 1(b). In this case, node A detects the transmission failure because no echo from node B is received till the retransmission timer exceeds. Node A sets the next available route to the packet header in advance of the expiration of the retransmission timer, and retransmits the packet at once if the maximum lifetime of the packet does not exceed. At the same time, the retransmission timer is initialized for the next retransmission. Figure 1(c) shows when a packet reaches its destination node. The destination no longer relays the packet and the previous node cannot get an echo. To delete the buffered packet in the previous node, the destination node creates an ACK packet exceptionally. In these examples, time slots synchronize at all terminals to explain the system easily. In real systems, they do not have to synchronize strictly because terminals

Table 1 Network Configuration Table

	Dest. Node 0	Dest. Node 1	...
Route 1	Route Info.	Route Info.	...
Route 2	Route Info.	Route Info.	...
⋮	⋮	⋮	⋮

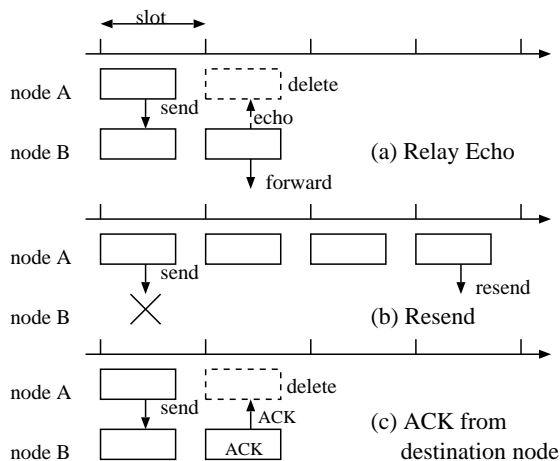


Fig. 1 Packet Transmission Timing

perform carrier sense before packet transmission and neighbors wait transmissions for one slot if they receive a carrier sense packet.

A maximum lifetime, that is a maximum time for a packet to be allowed to exist in the network, is pre-defined by slots for every data packet and is set at the source node. The lifetime is decreased by one for every timeslot even if it stays in a buffer. When the value reaches zero, the packet is discarded due to lifetime expiration. In other words, the packet is retransmitted repeatedly as long as the relay echo is not received and the lifetime of the packet remains. In the original FRN system, the value of this parameter is defined long enough for the network scale by users. However the maximum lifetime is an important configuration parameter, because short lifetime gives a chance to remove long-living packets effectively from the network while the long lifetime gives packets chances to try another route to the destination. Then, we measured the network performance changing the value of maximum lifetime in later simulations.

2.3 Routing Protocol

A routing protocol in ad hoc networks must select an appropriate route adaptively since the radio environment changes frequently. Furthermore, if a node fails to transmit a packet on the first trial, another route should be selected immediately (or transmission through the same route should be tried again). In the FRN, every node maintains multiple route information for each destination node in the configuration table as Table 1. Every node periodically broadcasts a network information packet consisting of the shortest route information to every node in the network. This is one-hop broadcasting and neighbor nodes do not relay the packet. When the neighbor node receives the information packet, it updates its network configuration table. This routing protocol targets at short-term errors since nodes are

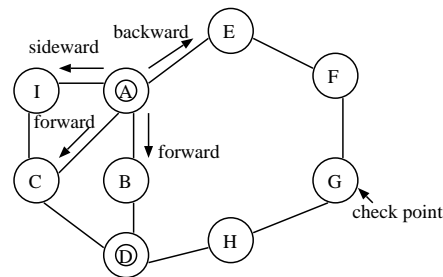


Fig. 2 Multipath Selection

stationary. Therefore, if once a transmission to the shortest route fails, the route is not deleted because the node expects the route will be available at the next transmission. If the number of network information packets received from a certain neighbor node becomes below a pre-defined threshold, the routes through the neighbor are deleted. We do not consider the delete of nodes in simulations. For each destination, routes are classified into three groups by their hop count:

- Forward route: The route(s) on which the hop counts to the destination is the shortest.
- Sideward route: The route(s) on which the hop counts to the destination is the shortest hop count plus one.
- Backward route: The route(s) on which the hop counts to the destination is the shortest plus two or more.

Figure 2 is an example of these route classifications. When node A is a source node and node D is a destination node, the shortest route between them has two hops, and the routes through node B and C are forward routes. The route through node I has three hops, so this route is classified for sideward route. There is a detour, a backward route, to node D that passes through node E. On a backward route, a checkpoint is defined for each destination at the source node to avoid back tracking. When the check point is defined in the packet header, nodes must transmit the packet to the checkpoint node at first. When node A transmits a packet through node E, node G is defined as the checkpoint. If not, node E will relay the packet back to node A because the shortest route to node D is through node A. We do not describe the detail of the checkpoint detection method in this paper.

In this routing protocol, the shorter route has the higher priority. When a packet transmission via the shortest route fails, the node cannot receive the relay echo and looks up its network configuration table again to select the second shortest route. In later simulations, the routing protocol selects another route after one trial to the higher priority route fails. In a routing operation, each node sets a next hop node ID to a packet header and broadcasts it. The neighbor nodes receiving this packet check the next hop field of the packet header

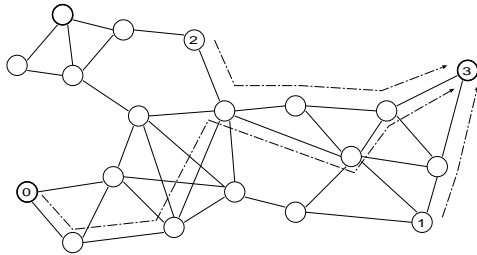


Fig. 3 Simulated Network Model

and recognize they are a next hop node or not of the packet. When they are, they decide new next hop of the packet and transmit it. When they are not, they check whether it is a relay echo (see Subsection 2.2).

3. Basic Properties of FRN

3.1 Simulation Environment

In this section, we first investigate basic properties of the FRN. We attend to the maximum lifetime by which the network performance is greatly affected as described in Subsection 2.2. We made some simulations with various values of the maximum lifetime and evaluated the correlation between the maximum lifetime and the system performance. For the simulations, we used the ns-2 [16] with its radio propagation model extended by CMU Monarch Project [17]. We used the multicast transmission mode of the MAC 802.11 for all packet transmission with a little modification to simulate time slot of the FRN. In all simulations, time slots synchronized at all terminals. This mode is a single hop multicast that does not produce the RTS/CTS/DATA/ACK exchange of the MAC 802.11 unicast mode. Radio transmission range was 250 m and buffer capacity of each node was 50 packets. It was large enough for inhibiting buffer overflow in our simulations. Every node broadcasted a network information packet at sufficiently long interval (at every 2500 slots). In addition, this was only one-hop broadcasting, hence the load of routing packets was much less than the data-traffic load. Therefore, they affect much less on the the network performance compared to the multi-hop data traffic.

We used a network model shown in Figure 3. A circle represents a node. A line connecting two nodes means that they can communicate directly. In this model, packet losses were assumed to occur only by the collision of the radio wave. The numbered nodes (node 0, 1, 2, 3) are host nodes that can transmit and receive data packets. An arrow from each source node to the destination is an example of the route that packets have actually passed through. In all simulations, source nodes went on sending constant bit rate UDP packets to node 3. The use of TCP is not concerned in this paper because the performance of FRN layers is

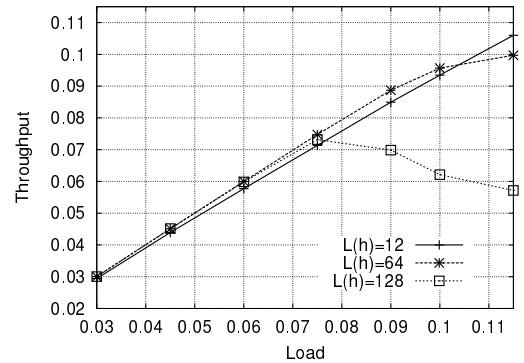


Fig. 4 Throughput in the Original FRN System

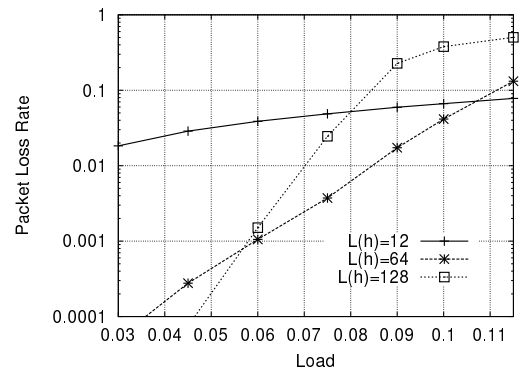


Fig. 5 Packet Loss Rate in the Original FRN System

measured well by UDP that does not equip the acknowledgement and the retransmission mechanism. This network model bases on the application of the FRN, collecting information from decentralized host nodes. The packet generation rate of each host node is assumed identical. The traffic load was defined as the number of packets generated per one slot in the whole network (i.e., the sum of the packet generation rate per one slot at three sender nodes). We simulated in the range of load from 0.03 to 0.115 which was practical in actual environment.

We used a throughput and a packet loss rate (PLR) to measure the performance. The throughput is an average number of packets transmitted to the destination successfully per one time slot. The PLR is a ratio of the packet not reaching the destination, that is, $1 - \text{PLR}$ equals to $\text{Throughput}/\text{Traffic load}$. In later simulations, we also evaluated the packet transmission time that was the average slots to reach the destination.

3.2 Network Performance for different Maximum Lifetime

We first describe the network performance with three different values of maximum lifetime, that are 12, 64, and 128 slots. Figure 4 shows throughput transition of each simulation. The label " $L(h)$ " indicates the

value of packet maximum lifetime. In long lifetime system, throughput results begin to degrade as the load increases because long-living packets cause congestion in the network. We can achieve high throughput in the high-load networks where packets have short lifetime value such as 12 slots. In contrast when the load is low, it is difficult to compare their results only by observing the throughput. It can be seen in the next Figure 5, the graph of PLR. When the load is low, the system applying long lifetime shows better performance than the one applying short lifetime. This is because the nodes can retransmit the packet after a transmission failure if the packet has a sufficiently long lifetime. Since every node in the FRN maintains multiple routes in its network configuration table, it can try the various routes on relaying packets to the destination when once a transmission fails.

Thus it is important to choose one appropriate value for the maximum lifetime. However it is difficult because it depends on the number and the density of nodes, and current load of the network. In the next section, we propose a method for calculating the adaptive lifetime of each packet based on the hop count to its destination node.

4. Performance Evaluation with Various System Parameters

4.1 Long Retransmission Interval

We described the hop-level packet retransmission mechanism of the FRN in Subsection 2.2. In the original FRN, nodes randomly select the length of the packet retransmission interval, which can range from 3 to 5 slots for each packet transmission. The relay echo system cannot work well when the retransmission interval is 2 slots because relayed packets collide with the next transmitted packet, while a longer interval does not cause such a problem. In this subsection, we investigate the effect of longer intervals on the system performance.

Let's denote the retransmission interval by **ret**. When the retransmission interval is selected randomly and it can range from 3 to 5 slots, we refer to the resulting system as a **ret=3-5** system. The retransmission interval length affects both the packet loss rate and the packet transmission delay. For example, a **ret=3-7** system will experience fewer packet collisions and longer transmission delays than the **ret=3-5** system. Let us now examine how longer retransmission intervals affect the system performance. We can increase the maximum lifetime of the packets in proportion to the average of all retransmission intervals, because all packets can experience retransmission fairly during their lifetime. For example, if the maximum lifetime is 12 slots in the **ret=3-5** system, it increases to 15 slots in the **ret=3-7** system and to 18 slots in the **ret=3-9** system.

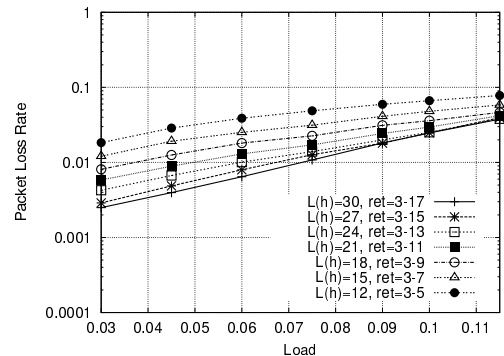


Fig. 6 Packet Loss Rate for Various Retransmission Intervals: $L(h)=12-30$

When **ret** is long, the probability of packet collision becomes low. However, end-to-end transmission delays may increase because the average waiting time for one retransmission increases. To investigate the effect of longer **ret**, we evaluated the loss rate and average transmission delay by simulations applying various retransmission intervals. The results for the throughput are not shown here because it is difficult to compare them in a low-load network as shown in Figure 4 where there are little difference among three maximum lifetime systems. Figures 6, 7, and 8 show that setting a long retransmission interval can effectively reduce the PLR when the maximum lifetime of the packets is long. This is because the probability of packet collision decreases with a longer retransmission interval. Of course, too long a retransmission interval will negatively affect the network. Figures 9, 10, and 11 show the average of end-to-end packet transmission delay when we change the value of **ret**. In these figures, the x-axis denotes the average length of the retransmission interval. When the maximum lifetime is short (Figures 9 and 10), the average transmission delay increases moderately as **ret** gets longer. This is natural because the number of packets staying long in the network increases. However, they do not seriously affect the network performance because the long-living packets are rejected when their short lifetime expires. In contrast, in long-lifetime high-load systems (Figure 11), the average packet transmission time decreases significantly as the value of **ret** gets longer. In high-load networks, packets repeatedly experience collisions, and take a long time to reach the destination node. Thus long retransmission interval decreases the probability of packet collisions and smoothen the traffic flow. Of course, packets must wait long time for a retransmission when **ret** is too long, therefore, there is a trade-off between the merit of decreasing packet collisions and the demerit of increasing the wait time for retransmission.

We used an analytical approach to explain this behavior. In the analyses, we assumed a string topology network and two host nodes at both ends. Pack-

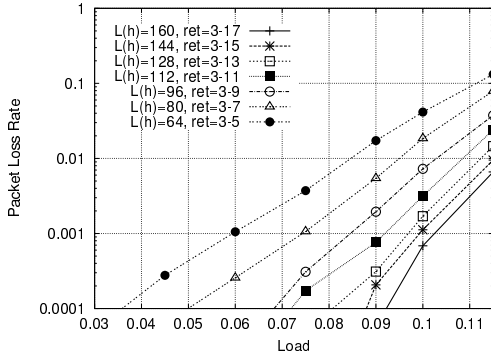


Fig. 7 Packet Loss Rate for Various Retransmission Intervals: $L(h)=64-160$

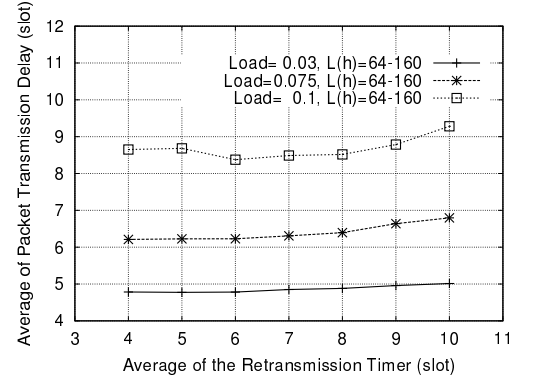


Fig. 10 Average Packet Transmission Delay for Various Retransmission Intervals: $L(h)=64-160$

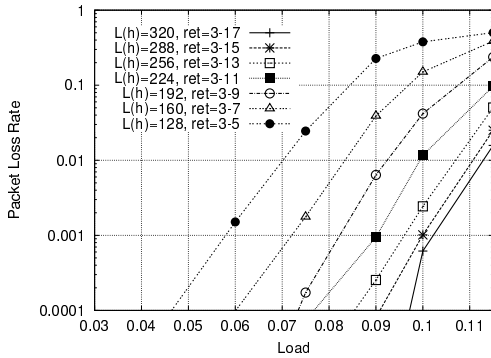


Fig. 8 Packet Loss Rate for Various Retransmission Intervals: $L(h)=128-320$

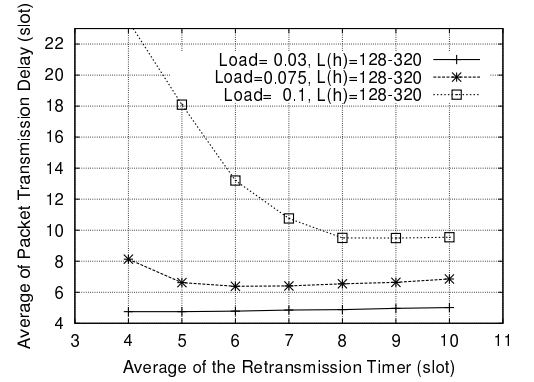


Fig. 11 Average Packet Transmission Delay for Various Retransmission Intervals: $L(h)=128-320$

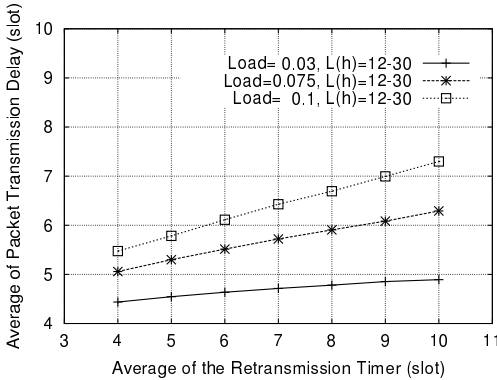


Fig. 9 Average Packet Transmission Delay for Various Retransmission Intervals: $L(h)=12-30$

ets transmitted between these two nodes as foreground traffic were assumed to have an infinite lifetime. Three other connections were generated as background traffic between intermediate nodes to simulate packet collisions. We defined P_s as the probability of collision of packets on its first attempt to be transmitted from a node. W was defined as the average waiting time for packet retransmission. In the $\text{ret}=3-5$ system, W was 4. If a packet collides with another packet, the probability, P_w , that it will also collide with another packet

at the next retransmission is given by:

$$\begin{aligned} P_w &= P_s + \frac{1}{2(W-3)+1} \\ &= P_s + \frac{1}{2W-5}. \end{aligned} \quad (1)$$

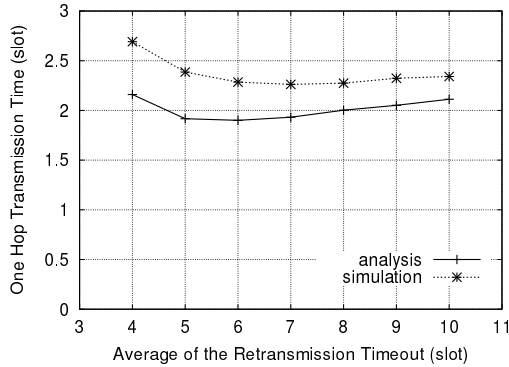
This is the sum of P_s and the probability that the retransmitted packets will collide again. Using these definitions, We can calculate $t(P_s, P_w, W)$, which is the expected time of one hop transmission:

$$\begin{aligned} t(P_s, P_w, W) &= (1 - P_s) + (1 + W)P_s(1 - P_w) \\ &\quad + (1 + 2W)P_sP_w(1 - P_w) + \dots \\ &= (1 - P_s) + \sum_{k=1}^{\infty} (1 + kW)P_sP_w^{k-1}(1 - P_w) \\ &= (1 - P_s) + \frac{P_s(1 + W - P_w)}{1 - P_w}. \end{aligned} \quad (2)$$

In the simulation, P_s strongly depended on the value of W . Table 2 shows the correlation of these variables measured on foreground traffic. Figure 12 shows

Table 2 Evaluated P_s for Various Values of W

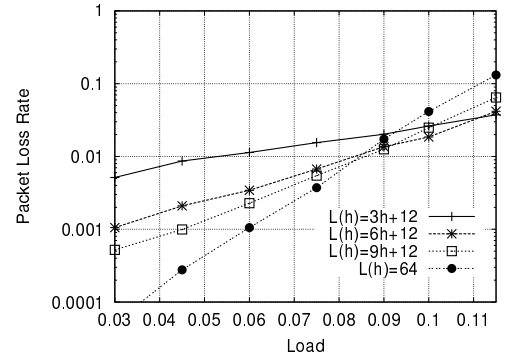
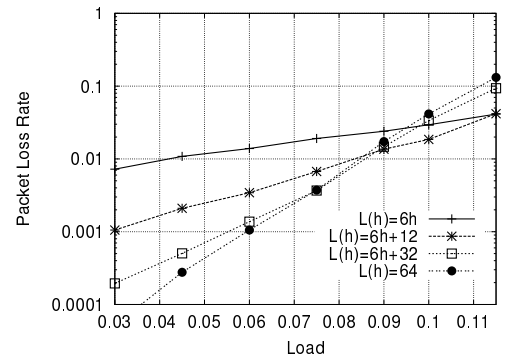
W	P_s
4	0.149848
5	0.123963
6	0.11187
7	0.104413
8	0.101181
9	0.09656
10	0.093426

**Fig. 12** One Hop Transmission Time: Analyses and Simulation

the simulated and analytical results of $t(P_s, P_w, W)$ obtained from Eqs. (1) and (2) and Table 2. The minimum time for one-hop transmission time was achieved when the average retransmission interval was seven slots ($\text{ret}=3\text{-}13$ system) in the simulation and six slots $\text{ret}=3\text{-}11$ system in the analyses. The difference between these two results was caused by our assumption in the analyses which did not take into account the buffering delay of packets in the intermediate nodes.

4.2 Adaptive Maximum Lifetime

The maximum lifetime of packets greatly affects the network performance (see also Section 3). In the original FRN system, all data packets had the same maximum lifetime, which was sufficiently long for the network. However, the lifetime required to transmit a packet to its destination varies for each packet because the traffic load in the network and/or the hop count on each route vary. According to the simulation results in Subsection 3.2, the maximum lifetime under 12 slots for this network is too short for the packets to be able to reach their destination. Many packets are dropped after several retransmission attempts even when the load of the network is comparatively low. In contrast, the lifetime exceeding 128 slots is too long in the high-load networks. The problem is that it is impossible to use the network load to determine the best value of the maximum lifetime because the load is the total of all the packet generation rates at every node, and the nodes cannot know the whole network load. In addition, packets go through different routes in the

**Fig. 13** Packet Loss Rate in the Adaptive Maximum Lifetime System (with various α)**Fig. 14** Packet Loss Rate in the Adaptive Maximum Lifetime System (with various β)

network, and the best value of the maximum lifetime is different for different packets.

We used the smallest hop count to the destination to calculate the necessary hop count for each packet. Generally, packets need a longer lifetime to pass through a longer route. It is easy to use the hop count to calculate the value of the maximum lifetime, because every node maintains the smallest hop count to each destination in a network configuration table. In this subsection, we describe the use of this method.

We formulated maximum lifetime L as a function of the shortest hop length, h . We used a simple linear function, $L(h) = \alpha h + \beta$, to calculate the adaptive maximum lifetime. Figures 13 and 14 show the PLR of setting various values of coefficients α and β , respectively on the function. The result obtained when the lifetime of the packets was fixed at 64 slots is also shown for the purpose of comparison. The simulation conditions were the same as in Subsection 3.1. According to these results, we found that setting of β to an appropriate value decreases the PLR in low-load environments, where the maximum lifetime of the packets transmitted on short routes must be sufficiently long in order to achieve a low PLR. Systems $L(h) = 6h + 32$ and $L(h) = 64$ showed almost the same performance in terms of the PLR. The difference was in the packet

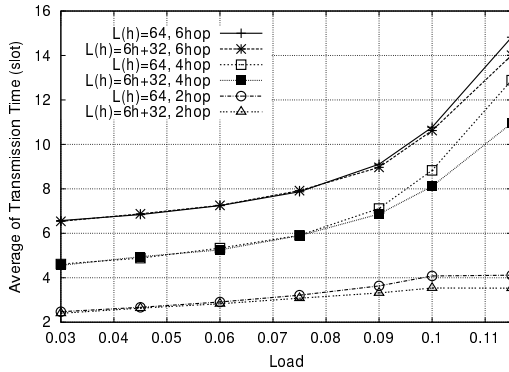


Fig. 15 Average Packet Transmission Time in the Adaptive Maximum Lifetime System

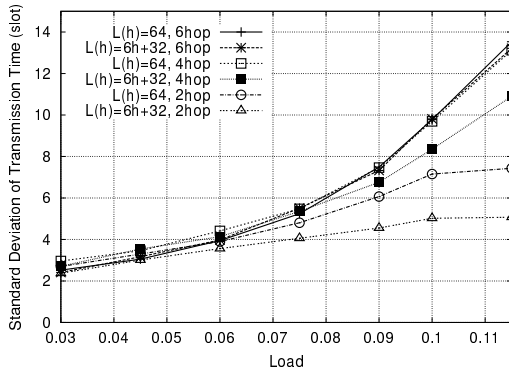


Fig. 16 Standard Deviation of Packet Transmission Time in the Adaptive Maximum Lifetime System

transmission time on the routes shown in Figures 15 and 16, which show the average and standard deviation of the packet transmission time for both systems. These results show that the adaptive maximum lifetime method achieves less average and fluctuation of transmission time especially in high-load networks than the static lifetime does even if there is little difference in terms of the PLR.

5. Packet Recording against Packet Duplication

As mentioned in Section 2, the FRN has a packet retransmission mechanism that ensures the system against transmission errors. While this mechanism is likely to improve the network reliability, it sometimes causes unnecessary packet duplication. Duplicated packets strongly degrade the network performance by increasing the network load and the number of packet collisions, and by occupying the node buffer. The packet duplication process is illustrated in Figure 17. Node A successfully transmits a packet to node B at slot 0. Node B relays the packet to node C at slot 1. This relayed packet should be received by node A as a relayed echo. However, sometimes the echo

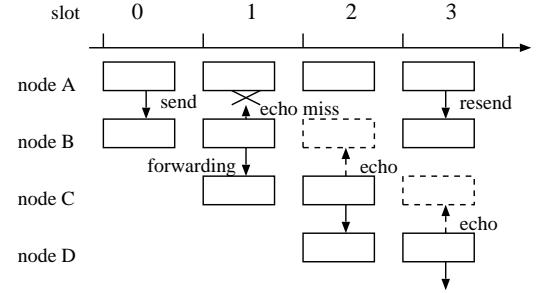


Fig. 17 Packet Duplication Process

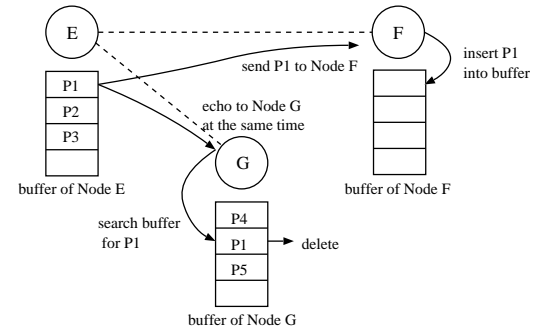


Fig. 18 Relay Stop by Eavesdropping

is lost because of a collision with another packet or a transmission error. Then node A cannot receive the echo successfully. In such a case, the packet left in the buffer of node A should not be removed at the end of slot 1. In other words, there are the same two packets in node A and node C. Node A retransmits the packet later via the same or another route to the destination, and as a result, it wastes the network resources.

In the FRN, there is a mechanism called *relay stop by eavesdropping*. This mechanism decreases the packet duplications. When the packet received by a node is the same as one of the packets in the node's buffer, the node erases the buffered packet. Figure 18 shows how a duplicated packet is rejected by this mechanism. We assume that Node E and Node G have the same packet named P1 in their buffers. In other words, one node has the original packet P1, and another has the duplicated one. Such situation can occur when the duplicated packet is transmitted through a route different from the original packet. Node E transmits packet named P1 to node F. Node G, the neighbor of node E, can also receive the packet. This is because all neighbors of node E can receive packets transmitted by node E wherever the packets are destined for, as described in Subsection 2.2. Node G examines its buffer and finds the same packet. It assumes that there is a duplicated packet in the same network, and erases the packet in its buffer. This mechanism can reject some duplicated packets; however, it cannot erase all duplicated packets and cannot prevent the duplication itself.

Therefore, we use a packet recording method to identify and delete duplicated packets more effectively. To identify duplicated packets, nodes record which packets have been successfully forwarded to the neighboring nodes when they receive an echo. They record the ID of each packet, created, for example, using the addresses of the source and destination, the sequence number, and so on. With these records, nodes can reject only the duplicated packets. See Figure 17 again. We assume that packet from node A is successfully relayed through node B to node C; however node A cannot receive the relay echo from node B. When node B receives the relay echo from node C at the next slot, it records the ID of the packet. When node B receives this retransmitted packet, it examines the packet's transmission record and determines whether the same packet has already been transmitted. As a result, node B can drop the packet retransmitted by node A. At the same slot, node B initiates a small acknowledgement packet destined for node A to delete the duplicated packet in the buffer of node A. The acknowledgement packet is the same packet as shown in Figure 1(c). Without this process, node A cannot delete the packet in its buffer and transmits the duplicated packets repeatedly.

However, sometimes an unduplicated packet, in other words, the original packet, may be dropped by mistake in this method. As described in Subsection 2.3 nodes in the FRN use the packet transmission mechanism via a backward route. If a node receives a packet already forwarded by the node, the packet will be dropped because the node has the transmission record of the packet. We used the route record in the header of FRN packets to solve this problem and to enable the nodes to differentiate between duplicated packets and original packets coming back to the node. A node may receive a packet that the node has already transmitted once and comes back to the node via a backward route. As a result, the original packet may be dropped in the packet recording method. The header area we used is called the node ID recording area. Every node records its node ID in this area before it transmits the packets. The node can identify a duplicated packet in its buffer, if it detects a packet with a successful transmission record but without a route record of the node ID in the packet header. In the simulations, the nodes did not drop the packets with the route record of itself.

Figure 19 shows that this method is more effective in decreasing the PLR when the packet maximum lifetime is longer. The lines labeled with "P_REC" indicate the performance of the packet recording system. As described above, the number of duplicated packets increases dramatically in a high-load environment. If the maximum lifetime is short, for example, 12 slots, retransmitted packets will be dropped because of their lifetime expiration, and this method will not be effective. On the other hand, packets with a long maximum lifetime cause more collisions and duplications.

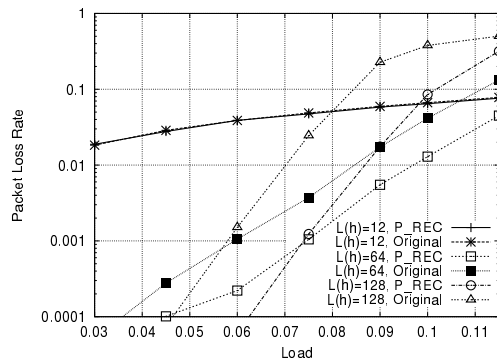


Fig. 19 Packet Loss Rate in the Packet Recording System

This unnecessary load on the network is reduced by the packet recording method, because it not only stops the transmission of duplicated packets but also defuses the source of packet duplication.

6. Conclusion

We investigated the performance of the Flexible Radio Network (FRN), a product for commercial use that is based on an ad hoc network system. Because of its application as a network, collecting information from many distributed terminals, the FRN adopts a proprietary protocol to construct a reliable network. However, its packet retransmission mechanism sometimes generates a high network load and frequent packet loss. To avoid collisions of retransmitted packets, we investigated the effects of longer retransmission intervals. We described an adaptive maximum lifetime setting technique to control the lifetime of the packets. We also described the packet duplication process in the FRN and proposed a method to identify and eliminate duplicated packets. We showed through simulation and analysis that these techniques can improve the network performance.

In the future, we should examine the end-to-end network performance when we apply, e.g., TCP, as an upper layer protocol. With such a bi-directional communication protocol, the packets will experience more collisions and the network performance will degrade. We also want to investigate the performance of a network with mobile terminals.


Acknowledgements

This work was partly supported by Special Coordination Funds for promoting Science and Technology of the Ministry of Education, Culture, Sports, Science and Technology of Japan.


References

- [1] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, "A feedback based scheme for improving TCP

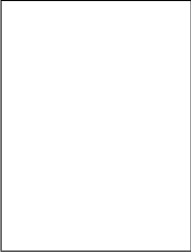
- performance in ad-hoc wireless networks," in *Proc. IEEE ICDCS '98*, (Amsterdam), pp. 472–479, May 1998.
- [2] T. Goff, J. Moronski, and D. S. Phatak, "Freeze-TCP – a true end-to-end TCP enhancement mechanism for mobile environments," in *Proc. IEEE INFOCOM 2000*, (Tel-Aviv), Mar. 2000.
 - [3] G. Holland and N. H. Vaidya, "Analysis of TCP performance over mobile ad hoc networks," in *Proc. ACM/IEEE MOBICOM '99*, (Seattle), pp. 219–230, Aug. 1999.
 - [4] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T. Chen, "Scalable routing strategies for ad hoc wireless networks," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1369–1379, Aug. 1999.
 - [5] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, "Scenario-based performance analysis of routing protocols for mobile ad-hoc networks," in *Proc. ACM/IEEE MOBICOM'99*, (Seattle), pp. 195–206, Aug. 1999.
 - [6] D. Kim, C.-K. Toh, and Y. Choi, "TCP-BuS : Improving TCP performance in wireless ad hoc networks," in *Proc. IEEE ICC 2000*, (New Orleans), June 2000.
 - [7] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," in *Proc. ACM/IEEE MobiCom 2000*, pp. 255–265, Aug. 2000.
 - [8] N. Nikaein, H. Labiod, and C. Bonnet, "DDR – distributed dynamic routing algorithm for mobile ad hoc networks," in *Proc. ACM/IEEE MobiHoc 2000*, Aug. 2000.
 - [9] M. R. Pearlman and Z. J. Haas, "Determining the optimal congruation for the zone routing protocol," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1395–1414, Aug. 1999.
 - [10] K. Takasugi, Y. Suzuki, and S. Kubota, "Multicast routing protocol for avoiding congestion in ad hoc wireless network," *IEICE Trans. Commun. (Japanese Edition)*, vol. J83-B, pp. 991–998, July 2000.
 - [11] "Flexible Radio Network, Fuji Electric Co., Ltd." available at http://www.fujielectric.co.jp/denki/p26/ecop_contents2.html.
 - [12] C. E. Perkins, *Ad Hoc Networking*. New York: Addison-Wesley, Dec. 2000.
 - [13] S.-J. Lee and M. Gerla, "AODV-BR: Backup routing in ad hoc networks," in *Proc. IEEE WCNC 2000*, Sept. 2000.
 - [14] S.-J. Lee and M. Gerla, "Split multipath routing with maximally disjoint paths in ad hoc networks," in *Proc. IEEE ICC 2001*, pp. 3201–3205, June 2001.
 - [15] M. Sugano, T. Araki, M. Murata, T. Hatauchi, and Y. Hosooka, "Performance evaluation of a wireless ad hoc network: Flexible Radio Network (FRN)," in *Proc. IEEE ICPWC 2000*, pp. 350–354, Dec. 2000.
 - [16] "The network simulator – ns-2." available at <http://www.isi.edu/nsnam/ns/>.
 - [17] "The CMU monarch project." available at <http://www.monarch.cs.cmu.edu/>.




Takayuki Yamamoto received the M.E. degree in Information and Computer Sciences from Osaka University, Japan, in 2002. He is now a doctoral student at the Graduate School of Information Science and Technology, Osaka University. His research interests include wireless ad hoc networks and their performance evaluation and simulation. He is a student member of IEICE.




Masashi Sugano received the B.E., M.E., and D.E. degrees in Information and Computer Sciences from Osaka University, Japan, in 1986, 1988, and 1993, respectively. In 1988, he joined Mita Industrial Co. Ltd. (currently, Kyocera Mita Corporation) as Researcher. Since September 1996, he had been Associate Professor in Osaka Prefecture College of Health Sciences. He moved to the Faculty of Comprehensive Rehabilitation, Osaka Prefecture College of Nursing in April 2003. His research interests include design and performance evaluation of computer communication networks, network reliability, and wireless network systems. He is a member of IEEE, ACM, IEICE and IPSJ.



Masayuki Murata received the M.E. and D.E. degrees in Information and Computer Sciences from Osaka University, Japan, in 1984 and 1988, respectively. In April 1984, he joined Tokyo Research Laboratory, IBM Japan, as a Researcher. From September 1987 to January 1989, he was an Assistant Professor with Computation Center, Osaka University. In February 1989, he moved to the Department of Information and Computer Sciences, Faculty of Engineering Science, Osaka University. From 1992 to 1999, he was an Associate Professor in the Graduate School of Engineering Science, Osaka University, and from April 1999, he has been a Professor of Osaka University. He moved to Advanced Networked Environment Division, Cybermedia Center, Osaka University in April 2000. He has more than two hundred papers of international and domestic journals and conferences. His research interests include computer communication networks, performance modeling and evaluation. He is a member of IEEE, ACM, The Internet Society, IEICE and IPSJ.



Takaaki Hatauchi was born in Hiroshima, Japan, in 1959. He received the B.E. degree from Kinki University in 1982. He joined Fuji Electric. His current research interests are communication protocols for wireless system.



Yohei Hosooka was born in Tochigi, Japan, in 1976. He received the B.E. degree in Faculty of Engineering from Utsunomiya University, Utsunomiya, Japan, in 1999. In 1999, he joined Fuji Electric in Japan, and he is a research engineer of wireless application. His current research interests are in wireless network communication architecture.