Master's Thesis

Title

A Study on Transport/Routing Protocols on Ad Hoc Networks for High-Speed Data Service

Supervisor

Professor Masayuki Murata

Author

Takayuki Yamamoto

February 13th, 2002

Department of Informatics and Mathematical Science

Graduate School of Engineering Science

Osaka University

Master's Thesis

A Study on Transport/Routing Protocols on Ad Hoc Networks
for High-Speed Data Service

Takayuki Yamamoto

Abstract

An ad hoc network is a distributed and self-organized network, where wireless terminals exchange network information with neighbor terminals and construct a multi-hop network. The topology of such a network changes frequently because of the unsettled wireless environment and/or terminal mobility. Consequently, appropriate route maintenance is necessary. In this thesis, we first focus on a commercially available ad hoc wireless network system, the Flexible Radio Network (FRN). This system uses a proprietary protocol that provides multiple-route management and hop-by-hop transmission acknowledgement. We discuss the system performance, which was evaluated through simulations regarding the data-link and routing protocol of the FRN to clarify the system's basic properties, and then explain some FRN problems that degrade the system performance. After that, we describe techniques to improve the performance. These techniques are aimed at improving the system parameter setting adaptively or at remedying the problems statically. Through simulation results, we show how the techniques improve system performance.

In the second part of this thesis, we investigate high-speed data communication services provided over ad hoc networks. Ad hoc networks are expected to be integrated with wired networks and many applications will operate over TCP/IP (Transmission Control Protocol/Internet Protocol), so many studies have been dedicated to improving the performance of TCP over ad hoc networks. However, most of these studies assumed a persistent TCP connection. This is clearly inadequate because many TCP connections are actually short-lived. For such short-lived connections, the routing latency in ad hoc networks is considerable. We describe a new routing protocol, the Low-latency Hybrid Routing protocol (LHR), that is suitable for a network in which many TCP connections are short-lived. Through simulations, we have compared the performance of LHR with that of existing routing protocols. The simulation results show that LHR can establish TCP connections more quickly than other routing protocols.

Keywords

Ad hoc wireless networks

High-speed data service

Transport protocol

Routing protocol

Short-lived TCP connections

Routing latency

Contents

List of Figures

List of Tables

1 Introduction

An ad hoc wireless network is a self-organized network consisting of wireless terminals that communicate with each other and exchange network information. These terminals are able to relay packets for another terminal, and can form a wide-area multi-hop wireless network. An ad hoc network needs neither a wired backbone network nor base stations, so network installation, expansion, and removal can be done quickly and easily. Such a wireless infrastructure is applicable to, for example, distributed computing, disaster recovery, and military operation. A sensor network, a data-collection system using many distributed sensors, is another good application of an ad hoc network. In such a network, the network topology changes frequently because of an unsettled wireless environment and user mobility. Accordingly, many studies have been dedicated to analyzing the characteristics of such networks and/or developing new routing methods (see, e.g., [1-6]).

Existing ad hoc routing schemes can be classified into three broad categories.

1. Proactive routing scheme: wireless terminals in the network maintain routing tables. They exchange tables with their neighbor terminals and periodically update them. In this way, routes to all terminals are pre-computed.

2. On-demand routing scheme: wireless terminals make a route query on demand when they have a packet for transmission. A terminal maintains only routes to terminals with which it is directly communicating.

3. Flooding scheme: all packets are broadcast.

We will describe the first two approaches in more detail. The Destination-Sequenced Distance Vector (DSDV) [7, 8] protocol is an example of a proactive routing protocol, and is based on the Distributed Bellman-Ford (DBF) algorithm [9]. Routes to all terminals are proactively calculated, so that the terminal can instantly determine the route for transmission. However, terminals must exchange their route tables more frequently in high mobility networks, which causes high routing operation overhead and degrades network performance. In addition, the table size and route propagation delay become larger as the number of terminals increases. Dynamic Source Routing (DSR) [10] is an example of an on-demand routing protocol. It finds a route to the destination by exchanging a broadcast route query packet and a unicast route reply packet. The query packet accumulates the terminal addresses along the route in its header, and the destination replies through the reverse route. Terminals have only to maintain the routes in present use. In a high-mobility or high-error network, the on-demand route search process causes high overhead while it can find a better quality path than proactive protocols. The route search latency is also considerable.

The Flexible Radio Network (FRN) is a commercially available ad hoc wireless network system [11]. A large-scale network with stationary terminals can be easily installed into existing facilities with the FRN. In addition, the network can be extended by simply adding a new radio terminal where needed. The FRN adopts a proprietary protocol that can efficiently adapt to terminal failures or a change in network configuration. The routing protocol is table-driven and based on the distance vector algorithm [9]. The difference between the FRN routing protocol and existing proactive ad hoc routing protocols like DSDV is that the FRN is capable of maintaining multiple routes to each destination and transmitting packets via a detour if the shortest route is temporarily unavailable. This is important in ad hoc networks with mobile and unstable communication environments. The basic performance of FRN has been investigated in [12], however, it is not entirely clear how system parameters affect performance in terms of, for example, the throughput and packet loss rate. In the current system, these parameters are determined through trial and error, but in order to clarify the applicability of this system, a more detailed evaluation is needed.

In this thesis, we describe three techniques that will improve FRN performance. These techniques resolve problems found through simulated performance evaluation and/or experience in an actual environment. One technique consists of a change in the system parameter setting. All FRN data packets have a parameter called *maximum lifetime.* This is used to discard long-lived packets wandering in the network. In an original system, a value large enough for the network (larger than the network diameter) is identically set for all data packets. However, the maximum lifetime of a packet is closely related to the hop count needed for the packet to reach the destination node. The system perform better if an adaptive lifetime is set for each packet.

The other two techniques decrease the number of packet collisions that lead to packet duplication. As explained later, the FRN has a problem in that packets are unnecessarily duplicated in packet relaying process. In the FRN, wireless terminals check for packet transmission errors at every hop. When a transmission error is detected, they retransmit the packet after selecting the next available route. This kind of error can be detected when the terminal does not receive a corresponding ACK signal from the neighbor terminal within a pre-specified time. The problem is that the packet sender terminal perceives a transmission error when the ACK packet is lost after the data packet is successfully transmitted. In such a situation, the terminal re-transmits the data packet even though the first packet was not lost. The re-transmitted packet in such a case is a duplicate packet. The unneeded duplicate packet enters the network because, to reduce the complexity of wireless terminals, the FRN has no terminals that manage the packet transmission history. Packet duplication thus needlessly increases the traffic load. To make matters worse, as the traffic load rises, more duplicated packets are generated because the ACK transmission error probability increases. Thus, the network performance rapidly deteriorates. The two techniques to prevent packet collisions and ACK losses will be discussed in Subsection 2.5.3.

Next, we will investigate the data communication services on ad hoc networks. Many applications operating on wired networks use TCP (Transmission Control Protocol). Ad hoc networks will probably be integrated with wired networks by using TCP/IP, and many researchers have studied the performance of TCP over ad hoc networks in [3, 13-17]. However, most of these studies assumed a persistent TCP connection (i.e., that there was an infinite amount of data to transmit which would cause the connection to remain open indefinitely), and the steady-state throughput values were examined. Since many TCP connections are short-lived in a real world, this approach is clearly inadequate. For example, the average size of Web documents at several Web servers has been reported to be about 10 [Kbytes] [18]. Furthermore, in a sensor network, a promising ad hoc network application, the amount of data carried by each connection is small and most of the TCP connections would be short-lived. Since TCP is an end-to-end communication protocol for wireless and wired terminals, its modification for applications to a sensor network will not permit adequate protocol migration. Instead, we should consider adopting a new ad hoc network routing protocol suitable for short-lived TCP connections.

To improve the performance of short-lived connections, we need to tackle the following problems, which are not addressed in the existing routing protocols;

- Large routing table overhead
- Large latency for the initial route search process
- Large latency for the another route search associated with link disconnection

If we assume a persistent TCP connection as in existing routing protocols, these problems will not much affect performance even in high-mobility and high-traffic-load environments. However, a TCP connection is unlikely to exist indefinitely in actual use, and most connections are short-lived. In this thesis, we thus propose a new routing protocol, the Low-latency Hybrid Routing protocol (LHR), which is aimed at decreasing the connection-establishment latency for TCP connections to overcome these problems.

This thesis is organized as follows. In Section 2, we begin by describing the Flexible Radio Network (FRN) system and evaluate the relationship between its performance and the system parameter setting. We examine the packet duplication process in detail, and we show that the network performance rapidly deteriorates as the number of duplicate packets rises. We then describe techniques to improve the performance with respect to packet duplication. A simulation has shown that these techniques can reduce the number of duplicate packets and improve network performance. In Section 3, we investigate a transport layer protocol suitable for ad hoc networks used for data communication services and propose a new routing protocol for short-lived TCP connections. Finally, we conclude this thesis and outline several remaining research topics in Section 4.

2 Performance of the Flexible Radio Network

2.1 System Description

The Flexible Radio Network (FRN) is a multi-hop wireless network system developed by the Fuji Electric Company. It is a data-collection system that has been used as a kind of sensor network. Early FRN applications have included power consumption data collection, usage information collection from ski-lift gates, and sales account collection and monitoring of vending machines.

In the FRN, every wireless terminal is called a *node*. Nodes with which a node can communicate directly are called *neighbor nodes*. Every node is able to select a route for a packet and relay the packet to a neighbor node. Specifically, a *host node* that generates and receives data packets and other nodes called *relay nodes* make up a multi-hop network. Every node maintains the network information in a *configuration table* that contains the routing information from the node to each destination node (Table 1). The routing information consists of a list of the neighbor nodes on the route to the destination node and the hop count of the route. Every node periodically exchanges a configuration control packet with the other nodes, and updates its configuration table according to the packets from neighbor nodes.

Table 1: Network Configuration Table

	Dest. Node 0	Dest. Node 1	$\ldots$
Route 1	Neighbor Info.	Neighbor Info.	$\ldots$
Route 2	Neighbor Info.	Neighbor Info.	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$

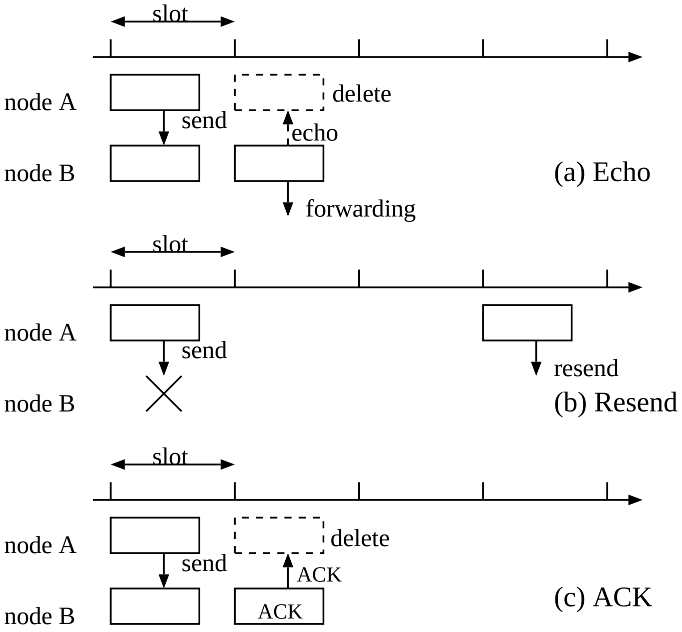

Figure 1: Packet Transmission Timing

2.2 Protocols Description

2.2.1 Data-link Protocol

In the FRN, hop-by-hop packet transmission acknowledgement ensures reliable packet transmission. A radio channel is divided into fixed-length time slots. In a wireless network, any neighbor node of a particular node can receive packets from the node even when it is not the source/destination of the packet. In the FRN, this property is used for the hop-level acknowledgment. Figure 1 shows an example of this. In Figure 1(a), the packet transmission and acknowledgment at node A is successful. Node B receives the packet from node A and successfully relays it to another node. At the same time, this relayed packet is received by node A because it is a neighbor node of node B. This acknowledgment is called the *relay echo* (or simply the *echo*). If the relay echo is successfully received by node A, it confirms that the transmission to node B was successful. In Figure 1(b), the first transmission from node A fails. In this case, node A detects a failure because it receives no echo from node B. Node A then retransmits the packet after a pre-specified time. When a packet reaches its destination node, the destination node transmits it no further and the previous node does not receive an echo. To delete the buffered packet in the previous node, the destination node creates an ACK packet as an exception (Figure 1(c)).

A maximum lifetime is predefined for every data packet. The lifetime is decreased by one per time slot even if the packet remains in a buffer, and the packet is discarded when the value reaches zero (its lifetime has expired). The maximum lifetime is an important configuration parameter since a short lifetime allows effective removal of long-lived packets from the network while a long lifetime gives a packet an extra chance to try another route to the destination. To establish a reliable network, we need to set the value high enough for the longest route, but low enough to avoid network congestion.

2.2.2 Routing Protocol

In the FRN, each node collects network information from its neighbor nodes and decides the direction for the relay of packets. Since the radio environment changes frequently, a routing protocol must adaptively select an appropriate route. Furthermore, if a node fails to transmit a packet on the first trial, another route should be selected immediately (or the same route should be tried again). In the FRN, every node maintains multiple sets of route information for each destination node in the configuration table as shown in Table 1. For each destination, routes are divided into three groups according to their hop count:

- Forward route: Route(s) with the lowest hop count(s) to the destination.
- Sideward route: Route(s) whose hop count to the destination equals the shortest hop count plus one.
- Backward route: Route(s) whose hop count to the destination equals the shortest hop count plus two or more.

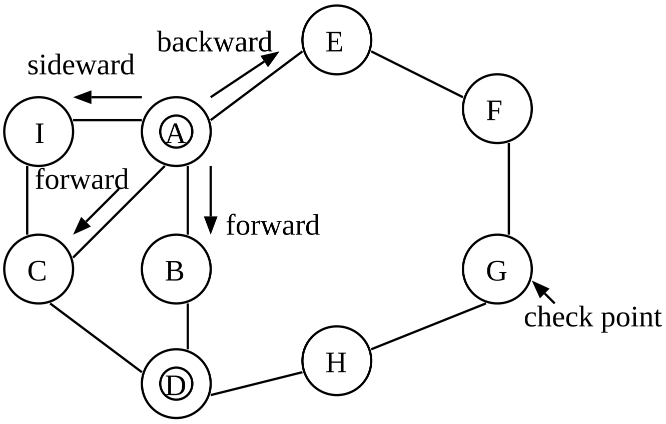

Figure 2: Multiple Route Selection

Figure 2 shows examples of these route classifications. When node A is the source node and node D is the destination node, the shortest hop count is two, and routes through nodes B and C are forward routes. The route through node I requires three hops, so this route is a sideward route. A backward route to node D 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 first transmit the packet to the checkpoint node. When node A transmits a packet through node E, node G is defined as the checkpoint. Without a checkpoint, node E would relay the packet back to node A because the shortest route to node D is through node A. (The checkpoint detection method is not discussed any further in this thesis.)

In this routing protocol, a shorter route has a higher priority. If the shortest route is unavailable for some reason such as an obstacle on the route, radio noise, or a packet collision, the node refers to the configuration table again and selects the second shortest route. Each node sets a next-hop node ID, a temporary destination, in a packet header after it decides the route. When neighbor nodes receive the packet, each neighbor node checks the next-hop field and recognizes whether it is a temporary destination node. If it is, the node decides a new next hop for the packet and transmits it. If not, the node checks whether the packet is the relay echo from its previous transmission.

2.3 Network Environment for Simulations

In this subsection, we describe the network envronment for simulations investigating the basic properties of the FRN. We focus on one system parameter – the maximum lifetime – that greatly affects the network performance. When the maximum lifetime is too long relative to the network diameter, some packets stay in the network for too long and degrade network performance. When it is too short, on the other hand, it may expire before relay nodes on the route can try another transmission route and the system cannot fulfill its potential. Through simulation experiments, we have evaluated a relationship between the lifetime parameter and system performance. For the simulations, we used the ns-2 network simulator [19] with a radio propagation model extended by the CMU Monarch Project [20]. Multicast transmission of the IEEE 802.11 [21] was used for all packet transmission. This mode is a single-hop multicast that does not produce the RTS/CTS/DATA/ACK exchanges of the IEEE 802.11 unicast mode, and is the same as the FRN at an abstract level, and thus sufficient to examine the FRN performance. The radio transmission range was 250 meters and the buffer capacity of each node was 50 packets. This capacity was large enough to inhibit buffer overflow, allowing the performance of data-link and routing protocols to be investigated. All the nodes periodically exchanged their network configuration tables. The period between exchanges was long enough, though, to prevent this affecting the system performance.

We used the network model shown in Figure 3. In the figure, a circle represents a node and a line connecting two nodes means they can communicate with each other directly. In this model, packet losses are assumed to be caused only by radio wave collision. The numbered nodes (nodes 0, 1, 2, 3) are host nodes that generate and absorb data packets. The dashed arrows from each source node to the destination are examples of routes that packets actually passed through. In all simulations, nodes 0, 1, and 2 continuously sent

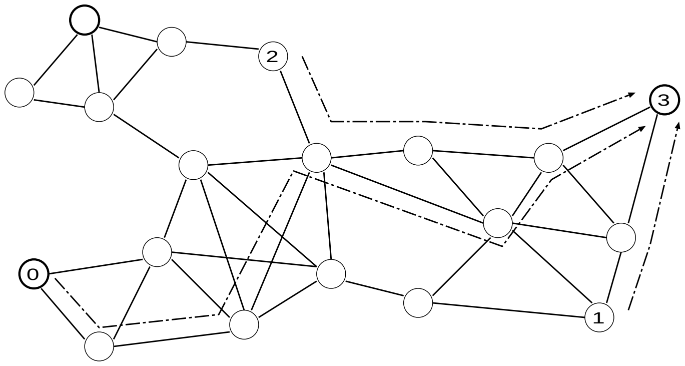

Figure 3: Simulated Network Model

UDP packets at a constant bit rate to node 3. This network model was based on an application collecting information from decentralized host nodes. An identical packet generation rate was assumed for each host node. The traffic load was defined as the number of packets generated per time slot over the whole network (i.e., the sum of the packet generation rate at the three sender nodes). We simulated loads ranging from 0.01 to 0.1, which are practical values in an actual environment.

The throughput and packet loss rate (PLR) were used to measure performance. Throughput was the average number of packets successfully transmitted to their destinations per time slot. The PLR was the proportion of the packets that did not reach their destination.

2.4 Performance Evaluation for Different Maximum Lifetime

The network performance is sensitive to the maximum lifetime as explained in the previous subsection. A long lifetime allows packets to be relayed to the destination node even after several retransmission attempt. As the traffic load rises, however, a lower value is an efficient way to drop long-lived packets and prevent congestion. There are three sets of simulation results discussed here, where the maximum lifetime value was set to 12, 64, and 128.

Figure 4 shows the throughput values. The label "L(h)" represents the maximum lifetime value. The load in the simulations was defined as the total of the packet generating

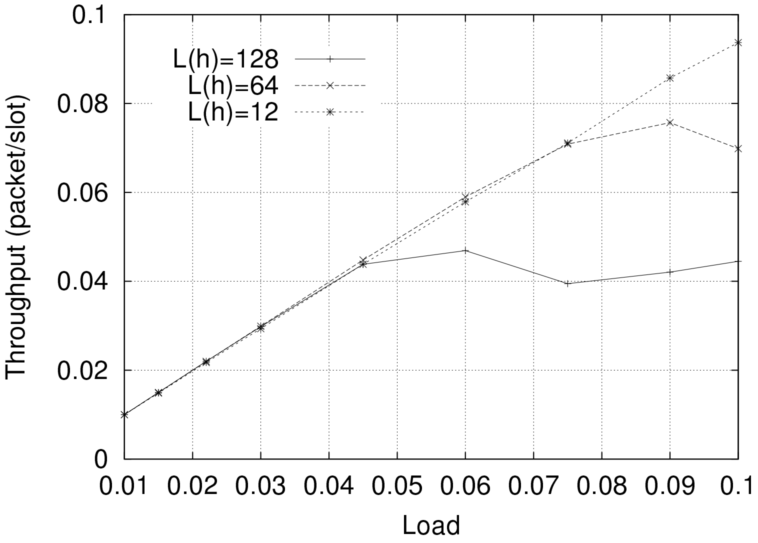

Figure 4: Original System Performance: Throughput

rate per slot for all host nodes. If the lifetime was long, throughput began to deteriorate as the load increased, because long-lived packets within the network caused congestion. We could achieve higher throughput in a high traffic load environment by using a lower lifetime value. However, the PLR exhibited a different tendency as shown in Figure 5. When the load was low, the system performed better with a long lifetime than with a short lifetime. As explained in Subsection 2.1, every node in the FRN maintains multiple routes in its configuration table. In the long lifetime system, nodes can try these various routes to relay packets to their destination once the initial transmission fails.

These results demonstrate the importance of selecting an adaptive value for the maximum lifetime. However, it is difficult to determine the best value because it depends on the scale, structure, current load, and traffic pattern of the network. A source node can only use the route information maintained in its network configuration table. Below, we explain how the adaptive maximum lifetime can be calculated by using the route length.

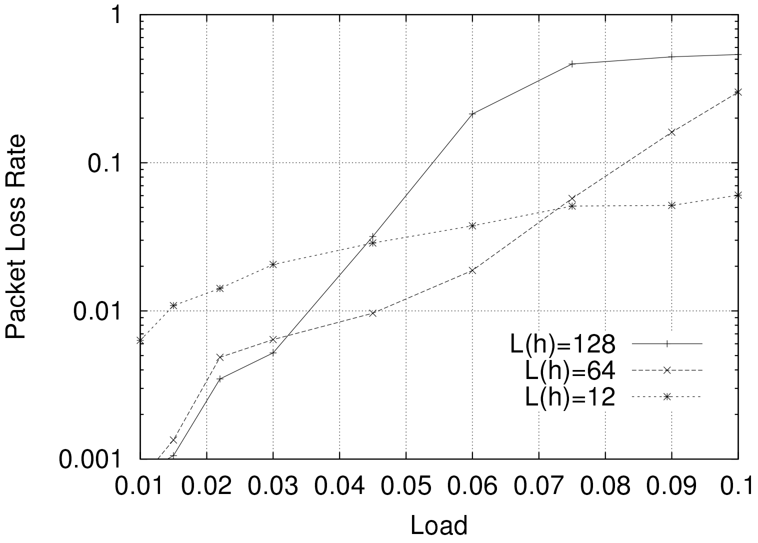

Figure 5: Original System Performance: PLR

2.5 Proposals for Performance Improvement

2.5.1 Adaptive Maximum Lifetime

The maximum lifetime value of the packet strongly affects network performance as described in Subsection 2.4. In the original FRN, all packets have the same maximum lifetime that is set according to the network scale. However, what actually constitutes an adequate lifetime for each packet differs for each packet because the hop count along the routes varies. In this subsection, we describe a method that can be used to set an adaptive maximum lifetime for each packet.

In the FRN, every node maintains a configuration table that includes the shortest hop path to a destination. This shortest hop count is closely related to the hop count necessary to reach a destination because a source node first tries the shortest route. When a packet is created at the source node, the node can dynamically calculate the necessary lifetime based on the shortest hop count in its configuration table.

The simulation results in Subsection 2.4 showed that a maximum lifetime under 12 is too small to decrease PLR when the load is low, while a lifetime over 64 is too large. We

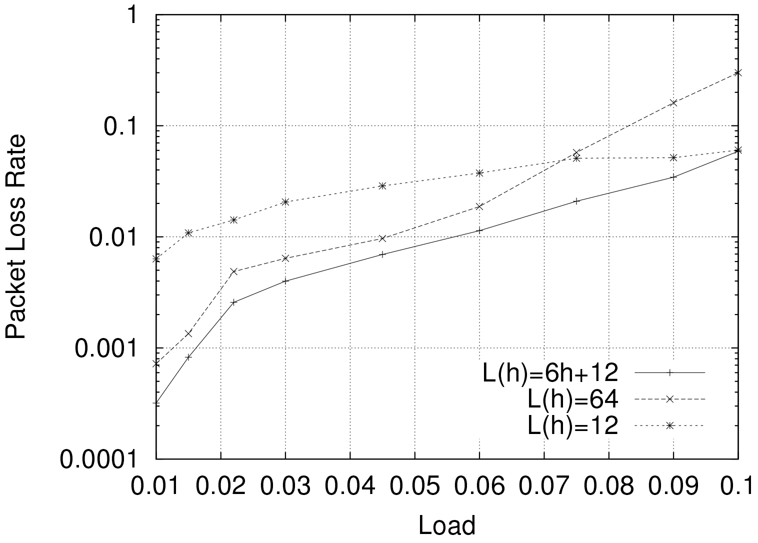

Figure 6: PLR for the Adaptive Maximum Lifetime System

can use, for example, $L(h) = 6h+12$ (the maximum lifetime L is a function of the shortest hop length h) as an adaptive maximum lifetime calculation. This function was derived through simulations. In a real environment, however, many factors will affect the necessary lifetime. Therefore, a unique function should be determined for each environment.

2.5.2 Performance Evaluation for Adaptive Maximum Lifetime

An example of the system performance when the adaptive maximum lifetime is used is shown in Figure 6. It is difficult to compare throughput values because there is little difference when the load is low, so only the PLR is shown. The PLR for static lifetime values of 12 and 64 are also shown for comparison. The simulation environment was the same as that described in Subsection 2.3. The function $L(h) = 6h + 12$ was applied to all packets at their source node to calculate their lifetime values. As shown in Figure 6, the modified system had a lower PLR than the system with static lifetime of 12 when the load was low. This was because the packets on longer routes could have a longer maximum lifetime in the modified system. When the load was high, the PLR deterioration was less rapid than with the long lifetime system. This indicated that the mechanism to discard long-lived packets according to their maximum lifetime worked better in the modified system than in the static long maximum lifetime system. Therefore, setting an adaptive lifetime for each packet is an effective way to control the trade-off between network congestion and reliability.

2.5.3 Fewer Packet Collisions and Duplications

As mentioned in Subsection 2.2.1, FRN has a packet retransmission mechanism to guard against transmission errors. While this mechanism improves packet reachability, it sometimes causes unnecessary packet duplication. A duplicate packet is a buffered packet in an intermediate node that will be sent out by the retransmission mechanism, while an original packet is being properly transmitted. Duplicate packets increase the possibility of collisions and occupy a node buffer. They increase the network load more than an original packet, and hence they can seriously degrade network performance. A packet duplication process is illustrated in Figure 7. Node A successfully transmits a packet to node B in slot 0. Node B relays the packet to node C in slot 1. At the same time, node A expects to receive the relayed packet as a relayed echo. However, the echo is sometimes lost because of an obstacle or a collision with another packet, and node A does not receive it. In such a case, the copy of the packet in the node A buffer will not be removed at the end of slot 1. In other words, the same packet exists in both node A and node C. Node A will then needlessly retransmit the packet, possibly through a different route to the destination because of the multiple route information in its configuration table.

Wireless terminals in the FRN have a feature called *relay stop by eavesdropping.* When a packet received by a node is the same as a packet in its buffer, the node stops relaying the buffered packet. Figure 8 shows an example of this feature. Node A transmits a packet (P1) to node B. At the same time, the packet can also be received by node C, the neighbor of node A, because all neighbors of node A can receive packets transmitted by node A regardless of the packet destination for, as described in Subsection 2.2.1. In other words,

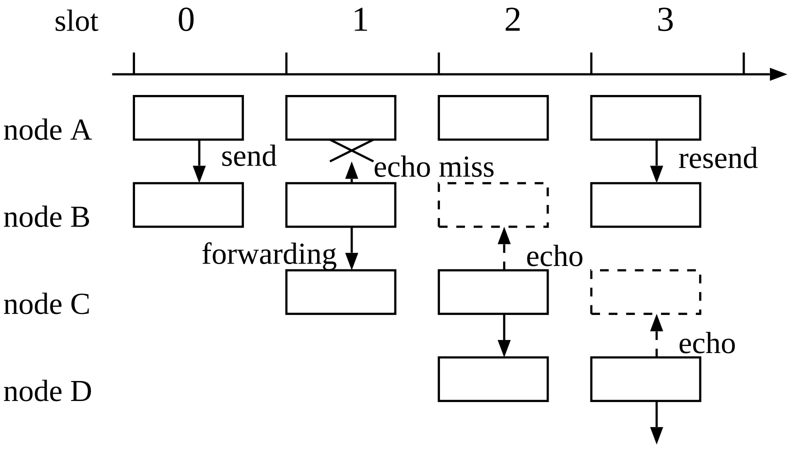

Figure 7: Packet Duplication Process

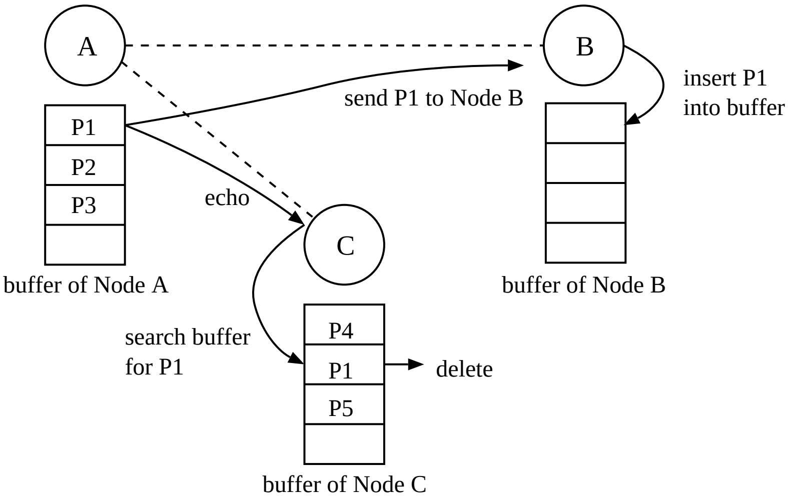

Figure 8: Relay stop by eavesdropping

node C can eavesdrop on the radio signal to node B. If node C refers to its buffer and finds the same packet, it concludes that there are duplicate packets in the same network, and so, discards the packet in its buffer. However, while this feature enables the system to discard some duplicat packets, it cannot erase all duplicat packets and cannot prevent the duplication itself.

Network performance falls rapidly as the number of packets in the network rises, as the results in Figure 5 clearly show. This is because the radio signal used for transmission

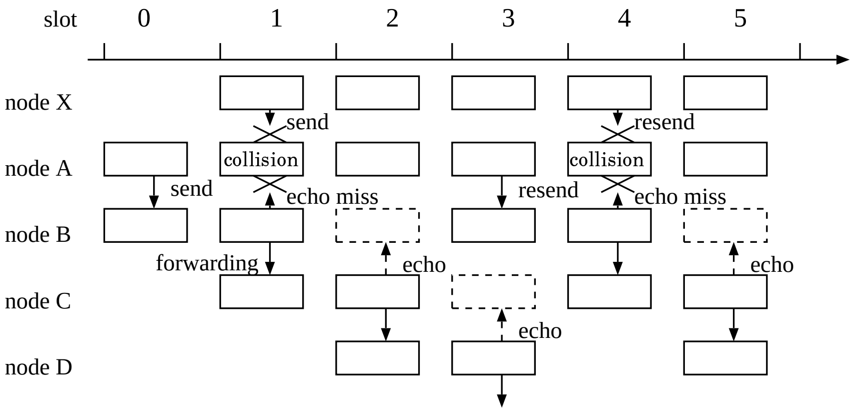

Figure 9: Repeated Echo Loss

affects all nodes within its range. Moreover, a duplicate packet is unnecessary if the original packet reaches its destination properly. Therefore, lowering the possibility of packet duplication will improve network performance. In this subsection, we describe two techniques that can appropriately control the number of duplicated packets. Since packet duplication is caused by loss of the relay echo, it can be reduced by (1) introducing a random delay before packet retransmission (RANDOM_DELAY), and (2) dropping packets that cannot reach the destination host within their remaining lifetime even through their lifetime has not expired (EARLY_DROP).

RANDOM_DELAY is an improvement of the packet retransmission method described in Subsection 2.2.1. This is a well-known technique to inhibit collision repeats. In the FRN, when a node does not receive an echo within a specified time, it selects another route for retransmission (Figure 1(b)). In the original system, this waiting period was constant (three slots) and it led to repeated collisions. Moreover, we found that in some cases, the relay echo packets repeatedly collide and many duplicate packets are produced. (Figure 9). Suppose that the waiting time before retransmission is fixed at three slots as in the original system. Node A expects to receive an echo from node B in slot 1. Node X, a neighbor node of A, transmits a packet in the same slot and this packet collides with the echo. Therefore, node A does not receive the echo and retransmits the packet in slot 3.

Next, in slot 4, the echo from node B collides again with the packet from node X because of the fixed waiting time. Nodes A and X each selects another route for retransmission if it has an alternative route entry to the destination, so it is difficult to detect the duplication on each node. Collisions are repeated until the transmission succeeds. RANDOM_DELAY prevents the repeated loss of relay echoes, thus reducing of the number of duplications. In the simulations, a random retransmission time from three to five was set for each packet.

EARLY_DROP is an effective method to lower the number of unneeded packets within the network. In the FRN, the remaining packet lifetime falls by one for every time slot that passes, even if the packet stays in a buffer. When no collisions occur, a packet can be passed one hop in each time slot. Ideally, the lifetime should be the maximum allowable hop count of the packet. The hop count to the destination is maintained at each node. Thus, if the residual lifetime of the packet becomes less than the hop count of the shortest path to the destination, that packet is of no use and should be immediately discarded. This technique prevents unnecessary packet transmissions and decreases the number of packet collisions that lead to packet duplication.

2.5.4 Performance Evaluation of Collision Prevention Methods

The results for RANDOM_DELAY modification are shown in Figures 10 and 11. The label "normal" denotes a result from the original system with no modification, and "random" denotes the RANDOM_DELAY modified system. With a long maximum lifetime (such as 64), the modified system is efficient because long lifetime packets can try several routes after a quick recovery from packet collisions. It worked as well, or only slightly worse, as when the maximum lifetime was short. In this simulation, nodes selected the waiting time before retransmission between time slots 3 and 5, and the average waiting time was longer than the fixed value of three in the original system. This technique sometimes caused more packet expirations than when the fixed lifetime was short, and thus is likely to be less useful in a short lifetime system. Figure 11 shows the number of duplications per

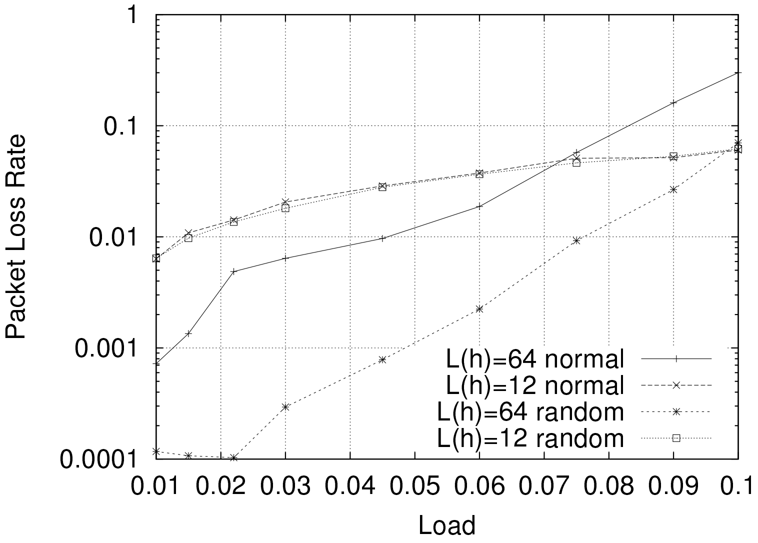

Figure 10: PLR for the RANDOM_DELAY Modified System

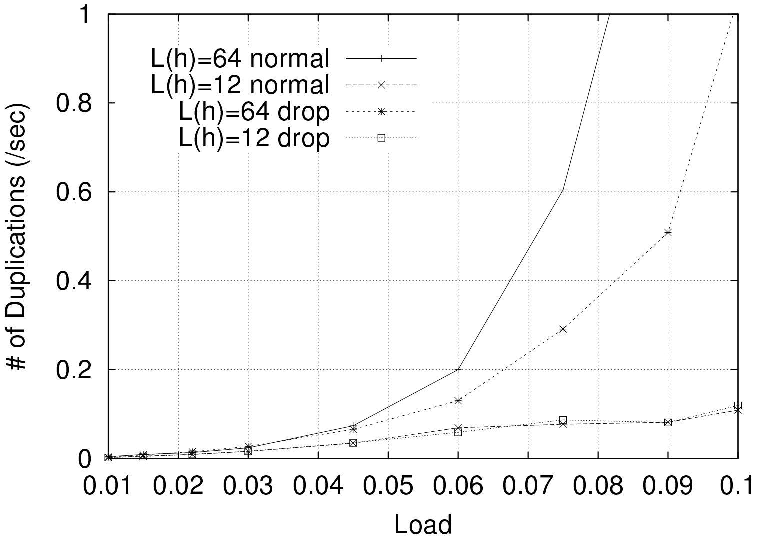

Figure 11: Number of Duplications for the RANDOM_DELAY Modified System

second. In a long maximum lifetime system used in a high traffic load environment, this technique can decrease the number of duplications and improve performance.

Figures 12 and 13 show the results for the EARLY_DROP modified system. (The results of the modified system are denoted by "drop" in the figures.) In contrast to the

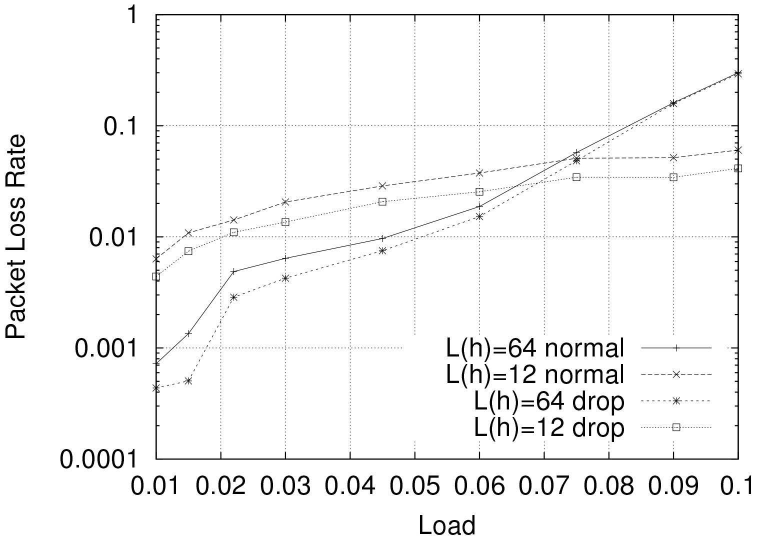

Figure 12: PLR for the EARLY_DROP Modified System

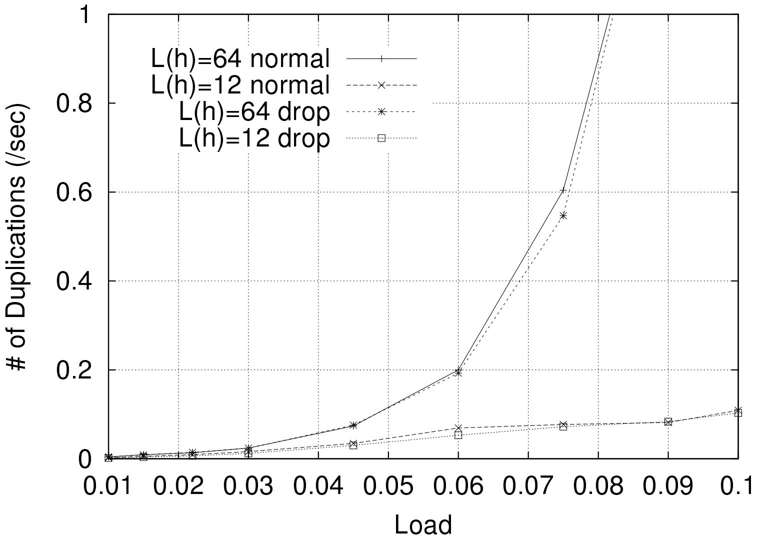

Figure 13: Number of Duplications for the EARLY_DROP Modified System

results with RANDOM_DELAY, the performance improved when the maximum lifetime was short. This modification had a beneficial effect on packets that would be dropped forcibly because of lifetime expiration. In the long lifetime system, packet timer expirations are less likely so this modification led to little improvement. That is, this modification does

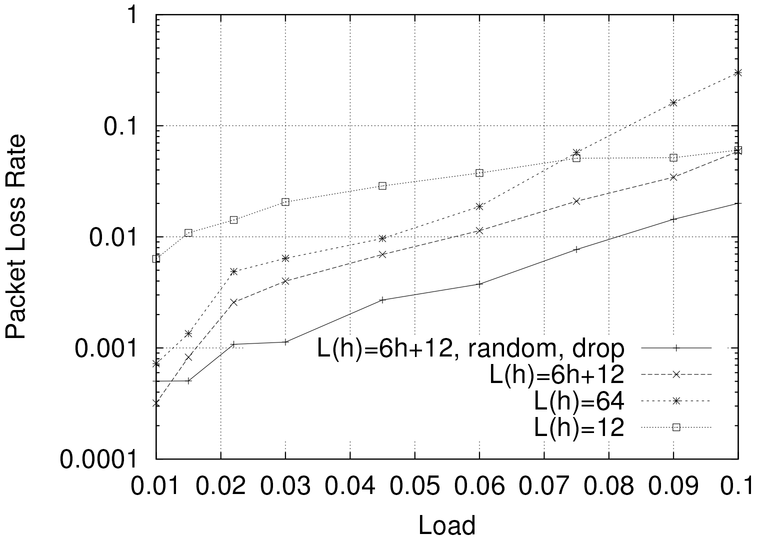

Figure 14: PLR for the All Modified System

not greatly lower the number of duplications because packets are quickly discarded because of lifetime expiration in a short lifetime system and little packet duplication occurs.

2.5.5 Performance Evaluation for All Methods Together

Now, let us consider the simulation results when using a method incorporating all three modifications; i.e., adaptive maximum lifetime, RANDOM_DELAY, and EARLY_DROP. We would expect the network performance to be further improved by applying all three at the same time compared to the improvement achieved by applying each separately. However, it was necessary to investigate their interaction because this could have an unexpected effect on performance improvement. Figure 14 shows the PLR dependence on the traffic load. The best performance was attained by applying all modifications. These simulation results confirmed that these techniques can improve FRN system performance.

3 End-to-End Communication in Ad Hoc Networks

The previous section described the hop-by-hop receipt acknowledgment method used to increase the network reliability. In the Internet, though, TCP is widely used as a reliable end-to-end protocol and its future use in wireless ad hoc networks appears certain. This section therefore focuses on the TCP performance in ad hoc networks.

3.1 Studies on the Transmission Control Protocol (TCP)

TCP was developed to ensure good performance in traditional wired networks where the probability of packet loss is expected to be low and most loss is caused by network congestion. A TCP source detects congestion if it does not receive an acknowledgement packet. In wireless networks, though, radio interference and node mobility raise the loss probability. However, the TCP source cannot determine the reason for an acknowledgement timeout. TCP-F [14] enables the TCP source to differentiate between timeout causes and to stop timers and transmission for as long as the source cannot contact the destination. TCP-BuS [3] also uses this idea and is capable of intelligent buffering techniques at mobile nodes. In [13, 15, 17], the TCP performance is compared in various network environments with respect to routing protocols, node mobility, and background traffic.

Most of these studies have assumed that TCP connections are persistent and have examined the steady-state performance. However as explained in Section 1, most of TCP connections are short-lived and the amount of data per connection is small in actual networks. For such connections, the routing latency in ad hoc networks will grow proportionately with the overall connection time and degrade network performance. Since TCP is an end-to-end communication protocol for both wireless and wired terminals, modifying it specifically for ad hoc networks will not allow adequate protocol migration. Instead, we need to consider a new routing protocol for ad hoc networks that is suitable for short-lived TCP connections.

3.2 Low-latency Routing Protocol for Short-lived TCP Connections

A sensor network is a typical application of wireless ad hoc networks. This network model, the collection of small amounts of information from many terminals, is suitably representative of the main characteristics of ad hoc networks, such as their distributed operation, scalability, and ease of maintenance. Many researchers have proposed various routing protocols for ad hoc networks, however, most of these protocols have been aimed at achieving good performance in a high-mobility, high-load environment and their effectiveness has been measured in a steady state. Specifically, they have assumed persistent connections in simulation experiments to evaluate protocols proposed in the literature. However, the characteristics of a practical network differ from those of the simplified networks that have been studied. For example, in a sensor network, (1) data is collected from many nodes and sent to a limited number of nodes; and, more importantly, (2) most TCP connections are short-lived.

To improve the performance of short-lived connections, we need to tackle the following problems, which are not resolved by the existing routing protocols;

- large overhead when exchanging the routing tables
- large latency of the an initial route search process
- large latency of another route search in the event of link disconnection

If we assume a TCP connection is persistent, these problems do not affect the performance even in a high-mobility, high traffic load environment. However, this assumption is not realistic.

We have developed a routing protocol that resolves the above problems and allows low latency short-lived connections. We call this the Low-latency Hybrid Routing protocol (LHR), and it combines on-demand route searching and proactive route maintenance. LHR can also reduce the latency of another route search in the event of link disconnection, as will be explained later in more detail.

3.2.1 Down-sizing Route Table

A sensor network collects data from many sensor nodes that generate data packets and transmit them to a few data-collection nodes. In such networks, the data-collection nodes are generally pre-specified and will not change when active. Routing protocols that maintain routes to all nodes will unnecessarily increase the network and terminal load. In LHR, each node registers the target destination nodes as *Data Receivers* (DRs). Only routes to DRs are maintained and exchanged with neighbor nodes. The method used to provide the nodes with the routes to the DRs is described in the next subsection.

3.2.2 Decreasing Latency of a New Route Search

On-demand protocols begin a route search upon packet transmission demand which tends to increase the latency. In contrast, proactive protocols can search for the route before the transmission demand actually occurs. However, proactive protocols mean a long time is needed to collect routing information from all over the network, and nodes cannot transmit packets to unknown destinations. In LHR, nodes maintain route tables for every DR and update the route entries with a periodic HELLO packet exchange. When they do not know an available route to the DR, they begin the route search process.

In LHR, a source node broadcasts a Route Request (RREQ) packet when it does not know any available routes to the destination. A node receiving a RREQ packet records the reverse route to the RREQ source node and rebroadcasts the packet if it is not the destination. The target destination receiving this packet broadcasts a Route Reply (RREP) packet. All nodes receiving these two packets register the target destination node as the DR and record routes. Then they begin to broadcast a HELLO message periodically to neighboring nodes, and update the routes to the DR. With proactive protocols, a link disconnection means that nodes must wait for a neighbors' route update message. To decrease this latency, LHR uses a different route re-search method, which is described in the next subsection.

3.2.3 Decreasing Latency of a Route Re-search

The most original point of our proposed protocol is the route re-search method. There are several techniques, as listed below, for recovering routes after link disconnection caused by node movement and/or changes in the wireless environment.

1. Routing tables are exchanged with neighbor nodes. Another available route can be found from the routing table. The problem is that there is no way to search for another route on demand.

2. A route error message to the source node is initiated. On receiving this message, the source node begins another route-request process. On-demand routing protocols such as DSR [10] use this method. It is effective for long-lived connections because the new route will be short and of good quality. However, in an environment with many short-lived connections, this approach seems to waste time.

3. A RREQ packet is broadcast from the node that detects the link failure. Though the route found with this method is sometimes longer than that with method 2, this does not seriously increase overhead when the connection time is short.

4. The network attempts to ensure that multiple routes are always maintained in advance. This means another available route is quickly provided. However, other available routes do not always exist.

We combined methods 3 and 4. In short, nodes suffering a link disconnection first try retransmission through method 4. If no other routes are available, the nodes try method 3 and will be able to recover the route within a short time. We describe these methods in more detail below.

As explained in Subsection 3.2.2, the node receiving a RREQ packet broadcasts a RREP packet. When a node receives the RREP packet from multiple neighbors, it indicates that there are multiple routes to the destination. The node then caches these routes

Table 2: Example of the Route Table of Node 0 in the Figure 15

	Neighbor Node	Hops	Sequence No.
route 1	4	2	1
route 2	3	3	1
route 3	2	3	1

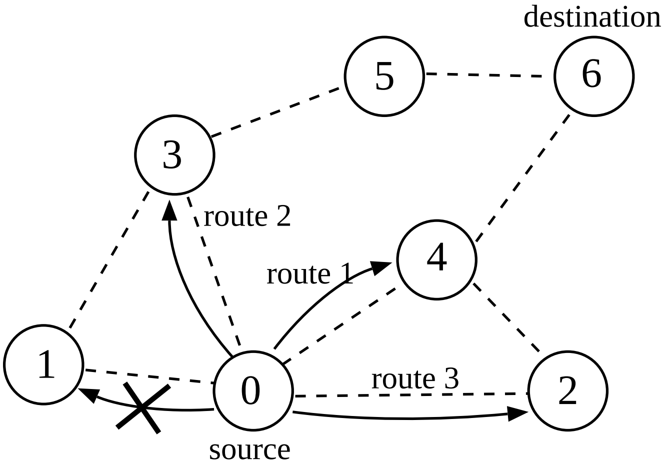

Figure 15: Multiple Routes' Entries

for use in the event of link disconnection.

When a node receives route information through a RREP packet from a neighbor node, it updates its routing table entries for the destination node specified in the packet. The node also compares the sequence number of the route information, as described in Subsection 3.2.2, with those of the route entries it maintains. If the sequence number is higher, the node deletes all older route entries to the destination node and begins to use the new route. If it is the same, the node can use that route as a backup route. If it is lower, the node simply ignores it.

With this multiple route maintenance mechanism, the number of route entries may become excessive. To avoid this problem, LHR limits the route entries for each active receiver. This limitation is that when the shortest route to an active receiver has n hops, the node maintains only n hop routes and $n+1$ hop routes. For example, in Figure 15, the shortest route from node 0 to node 6 has two hops. Node 0 will maintain only two-hop and three-hop routes. Table 2 is an example of the route table constructed in Node 0. It is difficult, though, to estimate an appropriate limit on the number of hop counts that the node should maintain. However, experience from past research into another ad hoc network systems [22] suggests that shorter routes should be given a higher priority. When node 0 has a packet destined for node 6, node 0 first tries the transmission on route 1, the shortest route to node 6. If it fails, node 0 disables route 1, and tries the second shortest route. If all transmission trials eventually fail, node 0 initiates a RREQ packet.

3.2.4 Integrated Connection Establishment

The TCP connection is established by a three-way handshake [23]. First, the TCP sender and receiver exchange SYN, SYN+ACK, and ACK packets. Because this negotiation is necessary regardless of the connection time, the time needed for connection establishment can be considerable relative to the length of short-lived TCP connections.

In LHR, two message packets are broadcast at TCP end-hosts when the route to a destination is unknown. Therefore, at the beginning of the TCP connection on a new route, they must exchange four packets before the source node receives SYN+ACK as illustrated in Figure 16. This creates a considerable latency for short-lived connections. We can decrease this latency when using LHR by integrating the TCP connection establishment with the route search (Figure 17). When the node initiating the SYN packet cannot find an available route, it broadcasts a RREQ packet carrying the SYN packet. The RREP packet also carries the SYN+ACK packet from the destination node. It is inevitable that the network load will increase, however, this is acceptable because we are now aiming at decreasing the latency for short-lived connections at the expense of an increased traffic load. Note also that this method does not make any changes to TCP, hence existing TCP

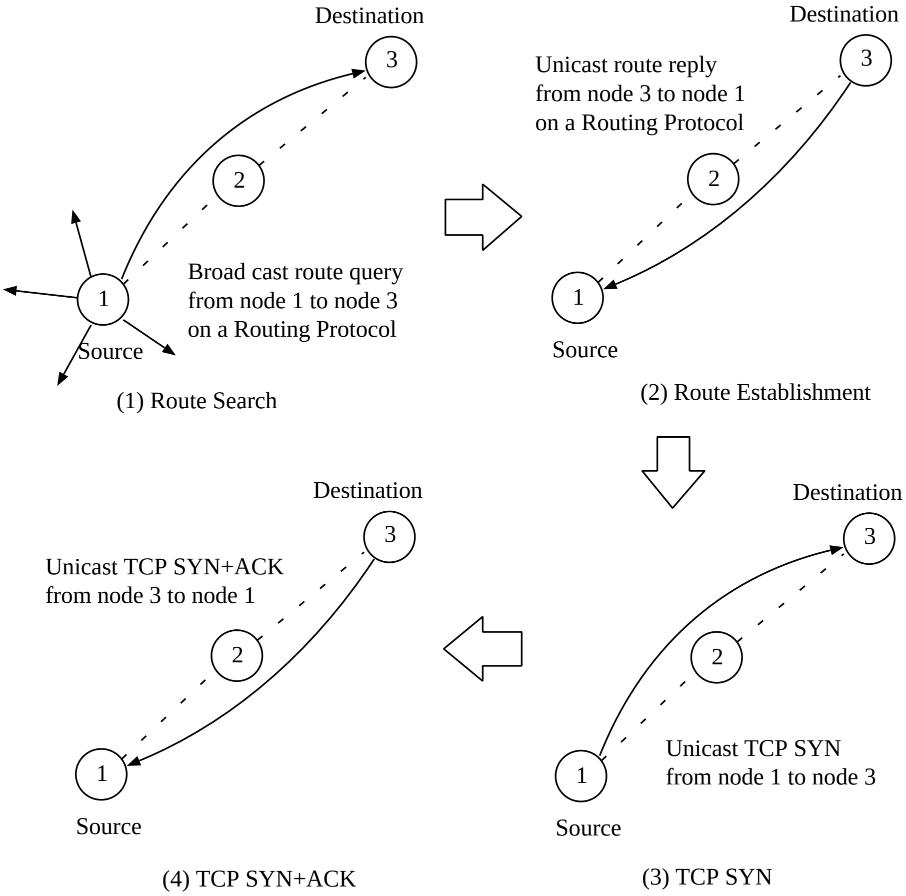

Figure 16: Sequential Operation of Route Searching and TCP Connection Establishment

implementations are applicable.

3.3 Performance Comparison

3.3.1 Simulation Methods

For the simulation experiments, we implemented LHR using an ns-2 network simulator [19] with a wireless extension developed by [20]. We used DSR and DSDV implementations by CMU Monarch [20]. An IEEE 802.11 wireless LAN [21] was employed at the link-layer level. The radio propagation range was 250 meters and the buffer capacity of each node was 50 packets.

We simulated a 500 x 2000 m network field that consisted of 50 randomly placed nodes.

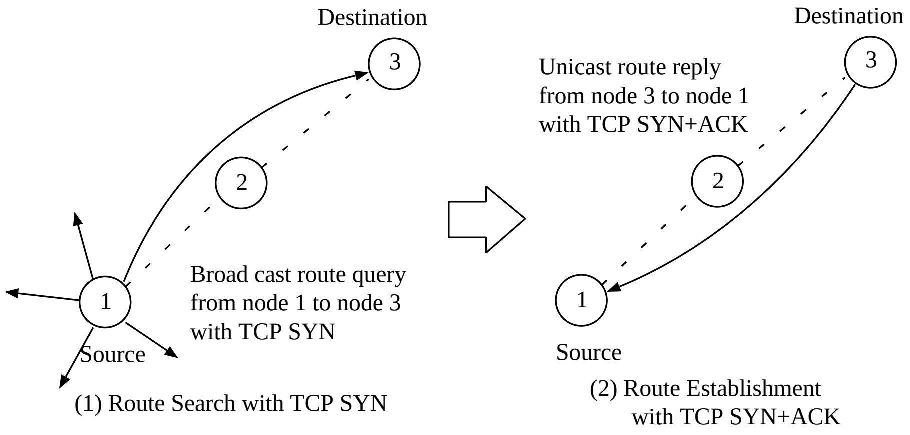

Figure 17: Concurrent Operation of Route Searching and TCP Connection Establishment

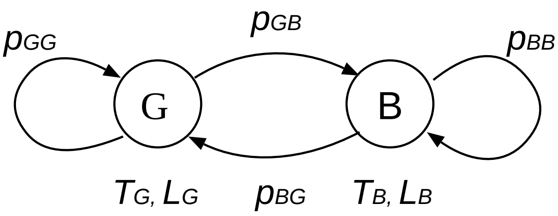

Figure 18: Two-state Transition Error Model

Their mobility pattern was based on a *random way-point* model [24]. To investigate the effect of node mobility, we used the three mobility patterns listed below.

- Max speed: 0 (m/sec)
- Max speed: 5 (m/sec), mean speed: 2.5 (m/sec), pause time: 0 (sec)
- Max speed: 20 (m/sec), mean speed: 10 (m/sec), pause time: 0 (sec)

Another important parameter that affects network performance is the packet error rate (PER). This was modeled by a two-state transition error model (Figure 18) known as the Gilbert model [25]. For each of the simulation experiments, we used two PLR values (0% and 1%), calculated from the expressions derived in [26], to evaluate how the PLR affects routing latency. We defined the unit time of good and bad states as the time needed to transmit one data packet. In our simulations, this was about 5.84 msec and was defined as one time slot. The mean length of staying in the error-free state was set to 20,000 slots, and that of the error state was 200 slots when we used an error rate of 1%.

We simulated a data collection network model with many short-lived TCP connections. In each connection, the amount of transmission data was uniformly set up to 10 packets and the TCP data packet size was 1,460 bytes. The simulation time was 210 sec and connection establishment processes were begun from 100 to 200 sec. During this first 100 seconds, DSDV nodes constructed the route tables. Connection establishment requests were assumed to follow a Poisson arrival process and the mean arrival rate in the network was set to 5 connections/sec. Connections were also assumed to be established to one data collecting node from all other data generating nodes. This traffic model was intended to simulate a data-collection network such as a sensor network.

We measured the TCP connection establishment delay for performance comparison. This delay is the time from TCP SYN generation to SYN+ACK receipt at the TCP source node. We also measured TCP data transmission time. This is the time between TCP SYN generation and the last ACK receipt.

3.3.2 Simulation Results

Figures 19 through 24 show the cumulative frequency distribution of the number of connections that could be established within the latency indicated on the horizontal axis.

As Figure 19 shows, in an error-free network with no mobility, all protocols (LHR, DSR, and DSDV) quickly establish most TCP connections. Among the three protocols, LHR enables the highest performance because all LHR nodes maintain routes to the DR (Data Receiver) and can establish the TCP connection in combination with the route search process. In this network model, no link disconnection occurs, so a proactive routing protocol such as DSDV can also quickly establish connection. However, if the number of link disconnections grows because of high node mobility, DSDV cannot establish connections

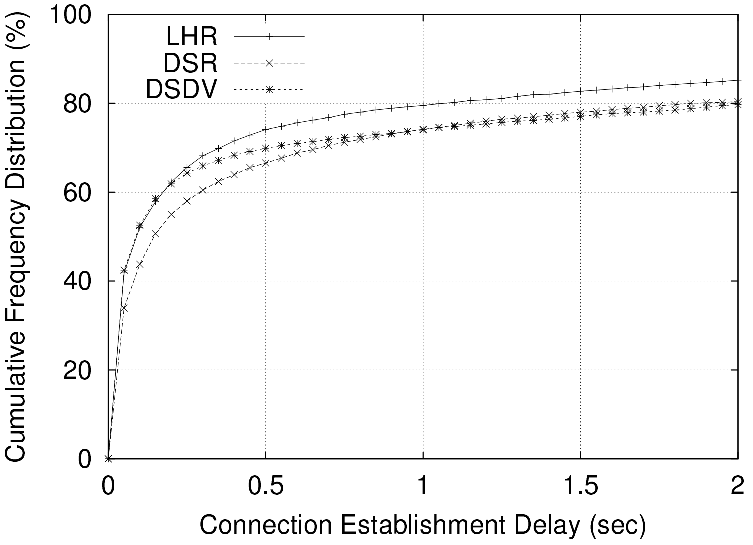

Figure 19: Connection Establishment Delay (Max speed 0 m/sec, PLR 0%)

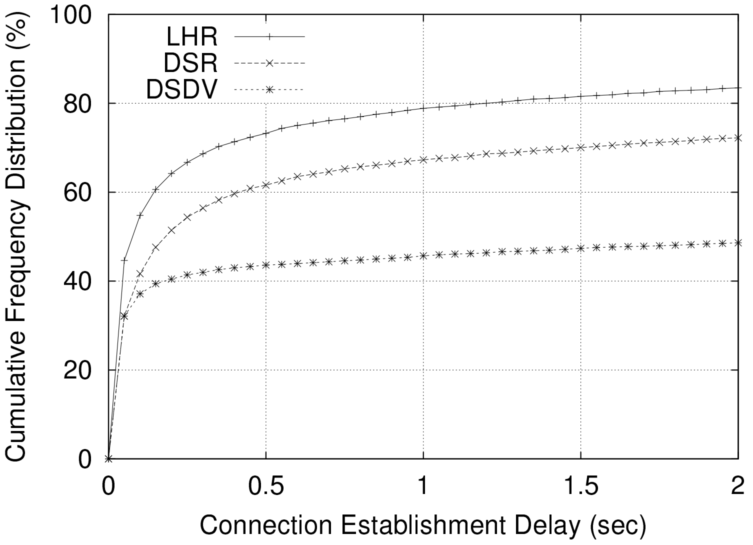

Figure 20: Connection Establishment Delay (Max speed 5 m/sec, PLR 0%)

as quickly (Figures 20 and 21), since proactive protocols are less able to accommodate the frequently changing topology of a high-mobility environment. While DSR can establish more connections than DSDV in a short time, LHR outperforms it.

When the PER value is set to 1%, the difference between protocols with and without

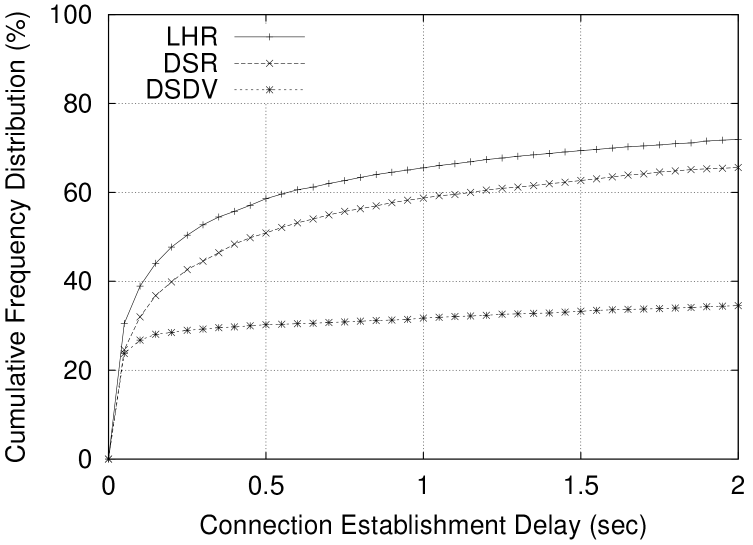

Figure 21: Connection Establishment Delay (Max speed 20 m/sec, PLR 0%)

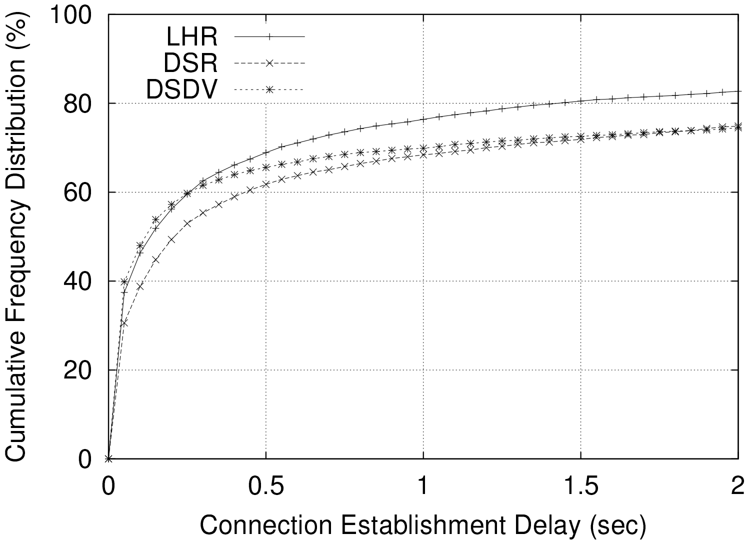

Figure 22: Connection Establishment Delay (Max speed 0 m/sec, PLR 1%)

an on-demand route search capability becomes increasingly prominent. As described in Subsection 3.3.1, we set the average length of the error state to 200 slots, and simulated burst errors. With DSDV, nodes cannot search for another route in response to such short and burst errors and they hold packets in their buffer until the route recovers. On-demand

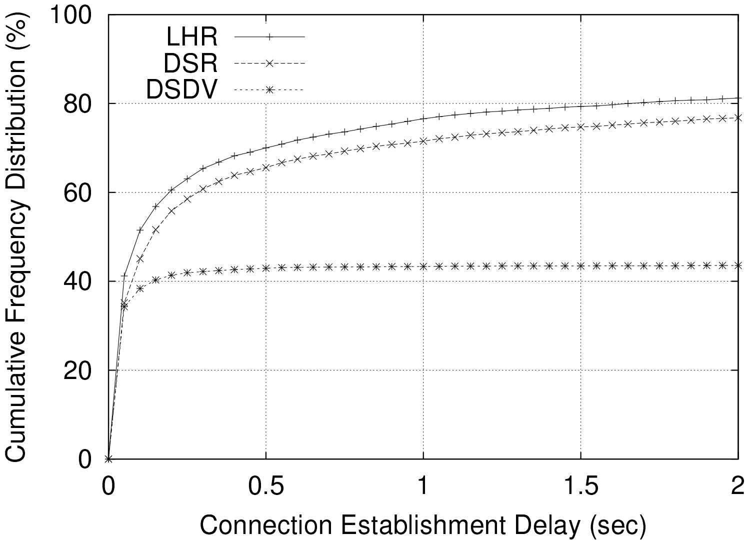

Figure 23: Connection Establishment Delay (Max speed 5 m/sec, PLR 1%)

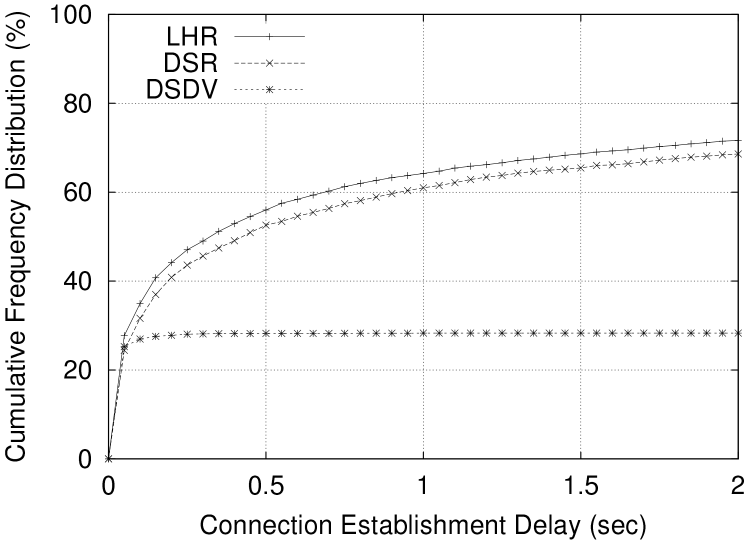

Figure 24: Connection Establishment Delay (Max speed 20 m/sec, PLR 1%)

protocols, in contrast, can search for another route to transmit packets. With LHR, nodes maintain multiple routes to each destination, so that they can select another route more quickly than with DSR.

Table 3 compares the connection establishment success rate. Each value in the table

Table 3: Connection Establishment Success Rate

Parameters	LHR	DSR	DSDV
Max speed 0 (m/sec), PLR 0%	98.7 %	97.8 %	98.2 %
Max speed 5 (m/sec), PLR 0%	97.9 %	98.1 %	77.3 %
Max speed 20 (m/sec), PLR 0%	99.0 %	97.6 %	63.9 %
Max speed 0 (m/sec), PLR 1%	98.7 %	97.8 %	98.2 %
Max speed 5 (m/sec), PLR 1%	97.8 %	98.0 %	69.3 %
Max speed 20 (m/sec), PLR 1%	98.7 %	97.6 %	54.0 %

exceeds the maximum rate in each of the previous figures, because some connections experienced a connection-establishment time-out due to route searching failure and/or packet interference. If a time-out occurs, the SYN packet is transmitted again, and after a certain number of trials, the connection will be established. The time-out length was longer than two seconds, hence the connections established after retransmission are not shown in the figures. With DSDV, the rate of successful connection establishment declined if the node mobility and/or PLR increased. This was because DSDV does not allow the recovery of adequate routes to the destination during the simulation time in a network with high mobility and a high error rate.

The cumulative frequency distribution of the data transmission time is shown in Figure 25. This distribution represents the results from the simulated network model with parameters of a 20 m/sec maximum node speed and a 1% error rate. It shows much the same pattern as the connection establishment delay results shown in Figure 24. Table 3 compares the proportion of connections whose data transmissions were completed within the simulation time. These results also indicate that a protocol without an on-demand route search capability cannot accommodate an unsettled network environment.

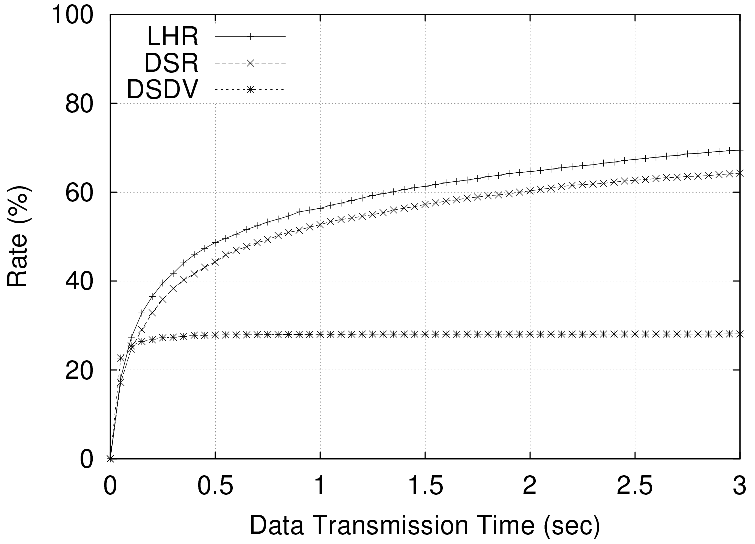

Figure 25: Data Transmission Time (Maximum speed 20 m/sec, PLR 1%)

Table 4: Completed Data Transmission Rate

Parameters	LHR	DSR	DSDV
Max speed 0 (m/sec), PLR 0%	98.3 %	95.9 %	97.2 %
Max speed 5 (m/sec), PLR 0%	96.6 %	94.8 %	75.3 %
Max speed 20 (m/sec), PLR 0%	96.9 %	94.6 %	62.8 %
Max speed 0 (m/sec), PLR 1%	97.7 %	94.3 %	95.6 %
Max speed 5 (m/sec), PLR 1%	95.7 %	95.3 %	67.3 %
Max speed 20 (m/sec), PLR 1%	97.1 %	95.2 %	53.4 %

4 Conclusion

In this thesis, we have investigated the routing and transport protocols used on ad hoc networks. We began by describing the Flexible Radio Network (FRN), a commercially available product, and evaluated its performance characteristics. Since its primary application is to collect information from many distributed terminals, the FRN uses proprietary protocols to construct a reliable network. However, these protocols sometimes cause a high packet loss rate because long-lived packets interfere with other packets, and sometimes make unnecessary duplicate packets that cause the network to become more congested. An adaptive maximum lifetime setting technique to control the lifetime of packets can reduce the number of long-lived packets. We also described the FRN packet duplication process and two techniques that reduce the negative effects of packet duplication. Simulations have shown that these three techniques can each improve network performance. In addition, they can be applied together to achieve even better performance.

Next, we have developed a routing protocol suitable for use in networks that have many short-lived TCP connections, e.g., sensor networks. In such networks, it is important to lower the connection and transmission latency for short-lived TCP connections. Our LHR (Low-latency Hybrid Routing) protocol enables on-demand route searching and proactive route updating. A data receiving node is registered as a data receiver, and all nodes maintain routes to the data receiver. To protect against the link disconnection due to wireless error and/or node mobility, nodes maintain multiple routes for each destination which decreases the route re-search latency. In addition, to decrease the initial connection establishment latency, an LHR route request and route reply packets can also carry the TCP connection establishment packets. Simulation experiments have shown that these features decrease the latency of connection establishment and enhance the system performance for short-lived TCP connections.

As part of our future research related to the FRN, we are now investigating the end-to-end performance at the time of applying, for example, TCP as an upper layer protocol. We also plan to develop a method for accurately determining an adaptive maximum lifetime while taking into account the network load and/or current packet loss rate. For LHR, we will experiment with other TCP versions or options (e.g., TCP SACK, delayed ACK). We will also investigate how the TCP window size and RTO calculation affect network performance.

Acknowledgements

I would like to express my sincere appreciation to Professor Masayuki Murata of Osaka University, who introduced me to the area of wireless networks including the subjects discussed in this thesis. He has also been a constant source of encouragement and advice throughout my studies and the preparation of this manuscript.

I would like to express my great gratitude to Associate Professor Masashi Sugano of Osaka Prefecture College of Health Sciences. None of the work discussed in this thesis would have been possible without him.

I also deeply appreciate the helpful advice of Mr. Takaaki Hatauchi, Mr. Yohei Hosooka, and Mr. Tomohiro Inayama of the Fuji Electric Company.

In addition, I am indebted to Assistant Professor Naoki Wakamiya and Research Associates Hiroyuki Ohsaki, Go Hasegawa, and Shin'ichi Arakawa of Osaka University who gave me many helpful comments and valuable feedback.

Finally, I thank my many friends and colleagues in the Department of Informatics and Mathematical Science of Osaka University for their support.

References

[1] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T. Chen, "Scalable Routing Strategies for Ad Hoc Wireless Networks," *IEEE Journal of Selected Areas in Communications*, vol. 17, pp. 1369–1379, August 1999.

[2] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, "Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks," in *Proceedings of ACM/IEEE MOBICOM '99*, pp. 195–206, August 1999.

[3] D. Kim, C.-K. Toh, and Y. Choi, "TCP-BuS : Improving TCP Performance in Wireless Ad Hoc Networks," in *Proceedings of IEEE ICC 2000*, June 2000.

[4] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," in *Proceedings of ACM/IEEE MOBICOM 2000*, pp. 255–265, August 2000.

[5] N. Nikaein, H. Labiod, and C. Bonnet, "DDR – Distributed Dynamic Routing Algorithm for Mobile Ad Hoc Networks," in *Proceedings of MobiHoc 2000*, August 2000.

[6] M. R. Pearlman and Z. J. Haas, "Determining the Optimal Conguration for the Zone Routing Protocol," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 1395–1414, August 1999.

[7] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance Vector Routing (DSDV) for Mobile Computers," in *Proceedings of ACM SIGCOMM '94*, pp. 234–244, 1994.

[8] C. E. Perkins, *AD HOC NETWORKING.* Addison-Wesley, 2001.

[9] D. Bertsekas and R. Gallager, *Data Networks.* Prentice-Hall, Englewood Cliffs, N.J., 1987.

[10] D. B. Johnson, D. A. Maltz, Y.-C. Hu, and J. G. Jetcheva, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," in *IETF Internet Draft.* `http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-06.txt`, November 2001.

[11] "Flexible Radio Network, Fuji Electric Co. Ltd." available at `http://www.fujielectric.co.jp/denki/p26/ecop_contents2.html`.

[12] M. Sugano, T. Araki, M. Murata, T. Hatauchi, and Y. Hosooka, "Performance Evaluation of a Wireless Ad Hoc Network: Flexible Radio Network (FRN)," in *Proceedings of the IEEE ICPWC 2000*, pp. 350–354, December 2000.

[13] A. Ahuja, S. Agarwal, J. P. Singh, and R. Shorey, "Performance of TCP over Different Routing Protocols in Mobile Ad-Hoc Networks," in *Proceedings of IEEE VTC 2000*, May 2000.

[14] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, "A Feedback Based Scheme For Improving TCP Performance In Ad-Hoc Wireless Networks," in *Proceedings of ICDCS '98*, pp. 472–479, May 1998.

[15] T. D. Dyer and R. V. Boppana, "A Comparison of TCP Performance over Three Routing Protocols for Mobile Ad Hoc Networks," in *Proceedings of MobiHoc 2001*, October 2001.

[16] T. Goff, J. Moronski, and D. S. Phatak, "Freeze-TCP – A true end-to-end TCP enhancement mechanism for mobile environments," in *Proceedings of INFOCOM 2000*, March 2000.

[17] G. Holland and N. H. Vaidya, "Analysis of TCP Performance over Mobile Ad Hoc Networks," in *Proceedings of ACM/IEEE MOBICOM '99*, pp. 219–230, August 1999.

[18] M. Nabe, M. Murata, and H. Miyahara, “Analysis and Modeling of World Wide Web Traffic for Capacity Dimensioning of Internet Access Lines,” *Performance Evaluation*, vol. 34, pp. 249–271, December 1999.

[19] “The Network Simulator - ns-2.” available at `http://www.isi.edu/nsnam/ns/`.

[20] “The CMU Monarch Project.” available at `http://www.monarch.cs.cmu.edu/`.

[21] IEEE Computer Society LAN/MAN Standards Committee, “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.” IEEE Std. 802.11-1997. IEEE, New York, NY, 1997.

[22] T. Yamamoto, M. Sugano, M. Murata, T. Hatauchi, and Y. Hosooka, “Performance Improvement in Ad hoc Wireless Networks with Consideration to Packet Duplication,” in *Proceedings of APCC 2001*, pp. 348–351, Septemper 2001.

[23] M. Naugle, *Illustrated TCP/IP*. Wiley, 1998.

[24] J. Broch, D. A. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva, “A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols,” in *Proceedings of MOBICOM ’98*, pp. 85–97, October 1998.

[25] E. N. Gilbert, “Capacity of a Burst-Noise Channel,” *Bell System Technical Journal*, vol. 39, pp. 1253–1265, September 1960.

[26] J. R. Yee and J. Edward J. Weldon, “Evaluation of the Performance of Error-Correcting Codes on a Gilbert Channel,” *IEEE Transactions on Communications*, vol. 43, pp. 2316–2323, August 1995.