PAPER *Special Section on Advances in Ad Hoc Mobile Communications and Networking*

A Low-latency Routing in Ad Hoc Networks for Many Short-lived Connections

Takayuki YAMAMOTO[†], ***Student Member***, **Masashi SUGANO**[††], ***and*** **Masayuki MURATA**[†], ***Members***

SUMMARY Wireless ad hoc network is expected to be integrated with wired networks and many applications will communicate over these networks transparently. Some applications need a reliable end-to-end transmission while wireless networks have less reliability than wired ones inherently. There are various ways to satisfy this demand. Many studies have been dedicated to improve the throughput of a connection over an ad hoc network. However, most of them have assumed persistent connections. This is clearly inadequate because major of connections are actually short-lived. For such connections, the routing latency in ad hoc networks is considerably long. In this paper, we propose a new routing protocol for ad hoc networks, the Low-latency Hybrid Routing protocol (LHR). LHR is designed to be suitable for an application where many devices need to transmit small data, while it is also applicable to mobile ad hoc networks. According to the simulation results, LHR is able to establish and process more connections within a given time period than other existing ad hoc routing protocols.

key words: *Ad hoc network, Low latency routing, Short-lived connection*

1. Introduction

Ad hoc wireless networks are self-organized networks built with wireless terminals. They communicate with each other and exchange network structure information. They can also relay data packets for another terminal to construct a wide area multi-hop wireless network. The ad hoc networks need neither a wired backbone network nor a base station. 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 analyse its characteristics and/or propose new routing methods (see, e.g., [1]–[6]).

Some applications in ad hoc networks such as a remote data logging need a reliable end-to-end transmission, although wireless ad hoc networks have less reliability than wired ones because of link error, packet collision, and loss of nodes. Therefore, a mechanism that a packet sender can know that the packet reaches its destination is necessary. The acknowledgement-and-retransmission mechanism of TCP in a transport layer is one possible solution. Many studies have considered the use of TCP over ad hoc networks (e.g., [7]–[9]). However, most of them assume that the TCP connection is persistent; i.e., it has an infinite amount of data to transmit, and then they examine a steady-state throughput. It is apparently inadequate because major of connections in many applications are short-lived. For example, it is reported in [10] that the average size of Web documents at several Web servers is about 10 [Kbytes]. Another approach to enhance the TCP performance over ad hoc networks has been proposed in [11], [12]. They have introduced a feedback-based mechanism into TCP. However, modifying TCP itself is not adequate for protocol migration since it is an end-to-end communication protocol including wireless and wired terminals. In this paper, we use TCP in all simulations to achieve a reliable packet transmission. We must note that TCP is not an exclusive solution. We consider that UDP is applicable as a transport layer protocol if an application layer protocol ensures the reliable end-to-end packet transmission.

In Fig. 1, we compare short- and long-lived TCP connections by separating the overall connection time into two blocks; one is a connection establishment time, which contains routing and TCP three-way handshake latency, and the other is a data transmission time. Comparing them, we can see that the connection establishment time is independent of the transmitting data size. In other words, the connection establishment time in a short-lived connection occupies larger percentage of overall connection time than that in a long-lived connection. When major of connections in a network are short-lived, we need to tackle the following problems, which are not resolved in the existing routing protocols;

- large overhead of exchanging the routing table
- large latency for an initial route search process
- large latency for another route search in the case of link disconnection

If we assume the connection is persistent, the above problems do not affect the performance even in high-mobility and high traffic load environment. However, connections are never persistent and most of them are

Manuscript received December 24, 2004.
Manuscript revised January 1, 2005.
Final manuscript received January 1, 2005.

[†]The authors are with Graduate School of Information Science and Technology, Osaka University, 1-5 Yamadaoka, Suita, Osaka 565-0871, Japan.
[††]The author is with Osaka Prefecture College of Nursing, Habikino-shi, 583-8555, Japan.

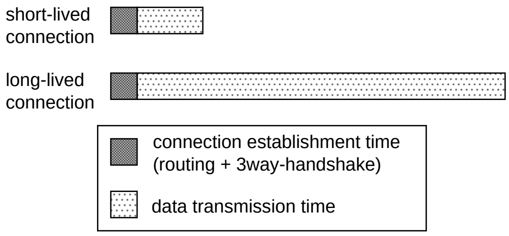


Fig. 1 Comparison of Connection Lengths of Short- and Long-lived TCP Connections

short-lived in real applications.

A sensor network is one promising application over wireless ad hoc networks. Small amount of data is collected from many terminals which equip sensors. Such a network model fits the most of characteristics and advantages of ad hoc networks, such as a distributed operation, scalability, and ease of maintenance. Sensors equipping a radio communication device can easily construct a wide area multihop network to gather their sensing data. Data traffic is routed to a destination node (i.e., data collecting terminal) by an ad hoc routing protocol.

There are two major routing methods, proactive routing and reactive routing, for ad hoc networks. Destination Sequenced Distance Vector (DSDV) [13] is one of proactive (table-driven) protocols that each wireless node exchanges a route table periodically with neighbor nodes. Ad hoc On-demand Distance Vector (AODV) [14] is a reactive (on-demand) protocol. A source node broadcasts a route query packet and all intermediate nodes create reverse route entries to the source node. The destination node receiving the query packet sends a route reply packet to the source node. In this paper, we propose a new routing protocol, Low-latency Hybrid Routing (LHR), which is suitable for an application such as a sensor network in which a few central node collect data from many distributed nodes and the amount of each data is small. Most of previously proposed protocols target to achieve high performance in high-mobility and/or high-load network, and they do not consider the traffic behavior of the upper layer protocols. LHR mainly targets at decreasing the routing latency by combining a reactive multiple route search and a proactive route maintenance. There are some advantages of LHR over the existing proactive and reactive hybrid routing like ADV [1]. LHR adopts a quick route re-search method against a link disconnection and also be able to combine a routing with a TCP connection establishment process to start data transmission quickly. T/TCP (TCP for Transactions) [15] equips a similar method to reduce the overhead of connection establishment. It integrates a TCP connection establishment process with data transmission. LHR can also work with T/TCP although we must consider that these methods fatten the broadcasted routing packets. As a result LHR can process more connections within a given time period. The remainder of this paper is organized as follows. We first describe the detail of LHR in Section 2. Simulation setup and results are shown in Section 3, and concluding remarks are made in Section 4.

2. Routing Protocol for Short-Lived TCP Connections

2.1 Decreasing the Overhead of Route Table Exchange

In some existing ad hoc routing protocols, more routes are maintained in a routing table than those actually used. One example is DSDV [13] that maintains routes to all nodes. However, such routing strategy unnecessarily increases the network/terminal load because it increases the size of route table with needless routes in current transmission requests. On the other hand, LHR registers the target destination node as an active data receiver like ADV [1]. Nodes maintain only routes to active data receivers and exchange them with neighbor nodes, so that the routing overhead can be much decreased compared to DSDV. While an initial connection to an inactive receiver takes large latency with a table driven routing protocol, LHR also adopts another on-demand route search mechanism to find a route to inactive receivers quickly. We describe it in the next subsection.

2.2 Decreasing Latency for New Route Search

LHR uses a reactive initial route search and a proactive route maintenance. Proactive routing protocols spend long time to collect routing information from all over the network, and nodes cannot transmit packets to unknown destinations. The interval of table exchange may be set long in energy-restricted environment. This cause much longer delay in propagating a correct route information. Then we consider that the proactive route search (including route re-search) is unsuitable for our protocol. On the other hand, they can find a route promptly if it is cached in the route table, hence, we consider that this method is suitable for the route maintenance.

In LHR, a source node broadcasts a Route-Request (RREQ) packet to search a route to an inactive receiver. The target destination node receiving the RREQ packet broadcasts a Route-Reply (RREP) packet. All nodes receiving these two packets register the target destination node as an active receiver. Therefore LHR is able to work with a network which contains simplex links. After nodes get one or more routes, they begin to send one-hop broadcasted HELLO messages periodically to neighboring nodes to update routes. It contains the destination and self node address, and a sequence number attached by the destination node which indicates the route freshness and is increased by one at every transmission of RREP. Some nodes that could not receive the RREQ or RREP packet register the routes by receiving the HELLO packet including a new route information. However, in case of the link disconnection, nodes must wait for the neighbors' route update message. To decrease this latency, LHR adopts another route re-search method, which will be described in the next subsection in detail.

2.3 Decreasing Latency for Route Re-search

In wireless ad hoc networks, nodes relay other nodes' packets. Some nodes and routes through it will be lost when the network is running because nodes are energy constrained and/or radio link is unstable. Route research method is an important factor of routing protocol. There are several techniques listed below for recovering routes against the link disconnection, which is caused by node movement and/or changes in the wireless environment.

1. The route is updated by exchanging a route table. Its problem is that it may take long time to get new available route.
2. A route error is acknowledged to the source node to make another route request. It would be effective for long-lived connection because the new route will be short and in good quality between endhosts. However, in an environment where there are many short-lived connections, this way apparently wastes time.
3. The RREQ packet is broadcasted from the node detecting the link disconnection. Though this method may make longer route than that of the above method 2, it is not a serious overhead when the connection time is short.
4. Multiple routes are always tried to be maintained beforehand.

We combined methods 3 and 4 to find another route quickly. Nodes suffering a link disconnection first try retransmission through method 4. If no other routes are available, they try method 3 and recover the route quickly. Next, we describe these methods in more detail.

In an initial route search, described in Subsection 2.2, a node may receive the identical RREP packets from two or more neighbor nodes. It indicates that there are multiple routes to the destination. The node caches these routes against the disconnection of the first route. This method can recover a route the most quickly in all ones listed above. When a node detects a link disconnection by feedback information from a data-link layer, the node inactivates routes through the link in its route table. The inactive routes are reactivated when a packet is received through the link again. Method 3 supports method 4. If the packet transmission through any cached route does not succeed, the node initiates a RREQ to find the current available routes to the destination.

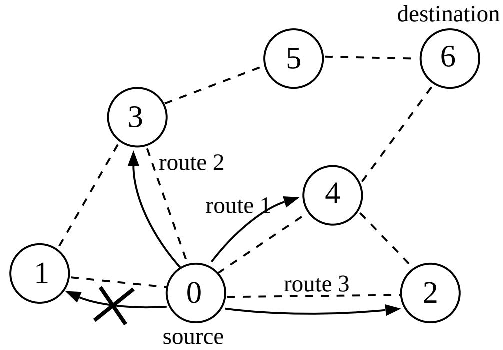
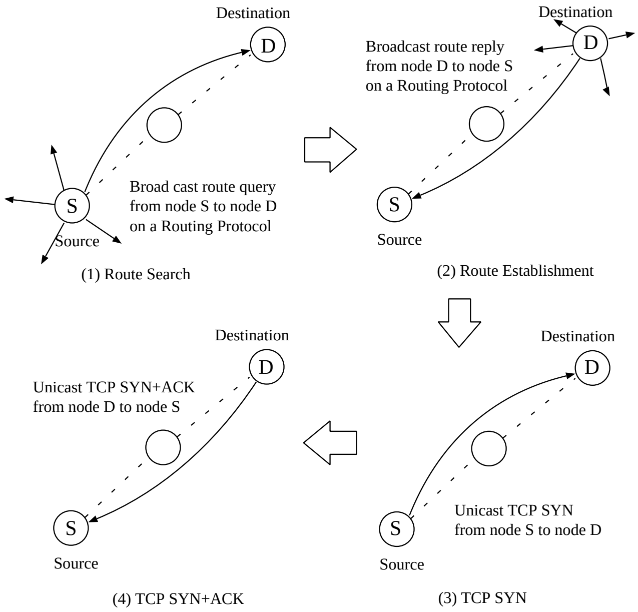


Fig. 2 Multiple Routes' Entry

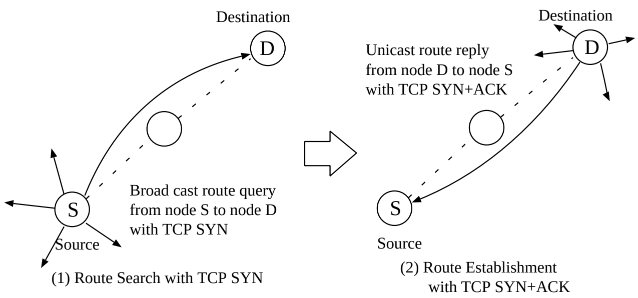
With this multiple routes maintenance mechanism, the number of route entries in the route table may increase too much. To avoid this problem, LHR adopts the limitation of route entries for each active receiver. This limitation is that when the shortest route to an active receiver has n hops, the node maintains only n hop routes and $n+1$ hop routes. See Fig. 2 as an example. The shortest route from node 0 to node 6 has two hops. The node 0 maintains only two hops and three hops routes. It is difficult to estimate the appropriate limit on the number of hop counts that the node maintains. However, we have some experiences from our past research about another ad hoc network system [16]. Based on the result in [16], the shorter route is 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 the transmission fails (i.e., the link layer detects the link disconnection), node 0 inactivates the route 1, and tries the second shortest route 2 or 3. If all transmission trials fail at last, node 0 initiates a RREQ packet to search a fresh route. This mechanism increases the number of routes maintained by each node. However, its affect is sufficiently small and is negligible. We will investigate the simulation results in Section 3 to see how many routes will be maintained by one LHR node.

2.4 TCP Connection Establishment Integrated with Routing Protocol

The TCP connection is established by three-way handshake. At first, TCP sender and receiver exchange SYN, SYN+ACK, and ACK packets. Because this negotiation is necessary regardless of the amount of data to transmit, the time for connection establishment becomes considerable especially in short-lived TCP connections. LHR processes TCP connection establish-



(a) Sequential Operation of Route Search and TCP Connection Establishment



(b) Route Search integrated with TCP Connection Establishment

Fig. 3 Connection Establishment Flow

ment and routing at one time. In [17], the usefulness of small data such as TCP SYN or reverse routing packet piggybacked by a routing packet was mentioned. However, our contribution is the implementation of the piggyback system to the protocol that is designed for a low-latency routing.

We explain how LHR combines the routing and the TCP connection establishment. As explained in Subsection 2.2, two message packets are broadcasted at TCP end-hosts to know a fresh route in LHR. Therefore, the end-hosts must exchange four packets (two routing packets and two TCP packets) till the source node receives the SYN+ACK packet if the source node does not know a route. It results a considerable latency for short-lived connections. We can decrease it by integrating TCP connection establishment with route search in LHR. See Fig. 3. When the node initiating the SYN packet finds no available route, it broadcasts the RREQ packet carrying the SYN packet together. The destination node records the source node address and port and store a RREP packet in a RREP waiting buffer. When TCP layer of the destination node hands a SYN+ACK packet, LHR checks its destination address and port. If there is a correspond RREP packet in the waiting buffer, LHR combines these packets and transmit it. Thus, the connection establishment time can be decreased. It is inevitable that the network load increases since the size of routing packets gains. However, it is acceptable because we now aim at decreasing the latency for short-lived connections at the expense of increased traffic load.

3. Simulation Experiments

We implemented LHR using an ns-2 network simulator [18]. We also used AODV and DSDV implementations of ns-2 for comparison. IEEE 802.11 Wireless LAN was employed at the link- and physical-layer. All routing packets of LHR was transmitted by the broadcast mode with setting the TTL to 1. This one-hop broadcast was used by other routing protocols in the same way. Data packets were transmitted by unicast mode with setting TTL (Time To Live) large enough (30). In all simulations, each node decreased the TTL value of a packet by one in relaying the packet in the same way of IP routers in the Internet. A feedback information from the link layer was employed to detect a link disconnection. The link layer retransmitted a data frame 7 times before it informed the link disconnection to the routing layer. We simulated a 500 x 2000 m^2 network field. Radio propagation range was 250 meters and the buffer capacity of each node was 50 packets, which was sufficiently large to prevent a packet loss due to buffer overflow. Packets were dropped only when the routing protocol could not find a route to the destination. Next, we explain why a routing protocol sometimes fails a route search. We assumed link disconnections among nodes were caused mainly by two reasons, a node mobility and a wireless error. The mobility pattern of nodes was based on a *random way-point* model [2]. To investigate the effect of node mobility, we used the two mobility patterns listed below. In our main target application, information collection from many wireless nodes, nodes may not move so fast and frequently. However we simulated also a high-mobility situation to show that LHR was able to work in mobile ad hoc network and had a better performance.

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

A wireless error was modelled by a two-state transition error model (Fig. 4) known as Gilbert model [19]. The

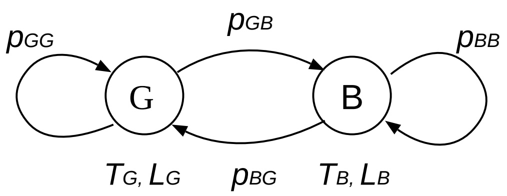
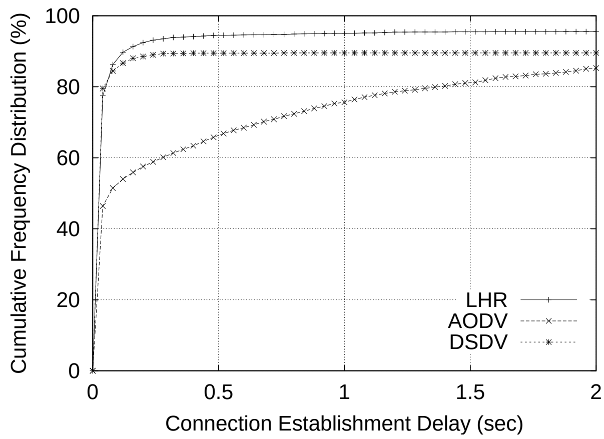


Fig. 4 Two-state Transition Error Model

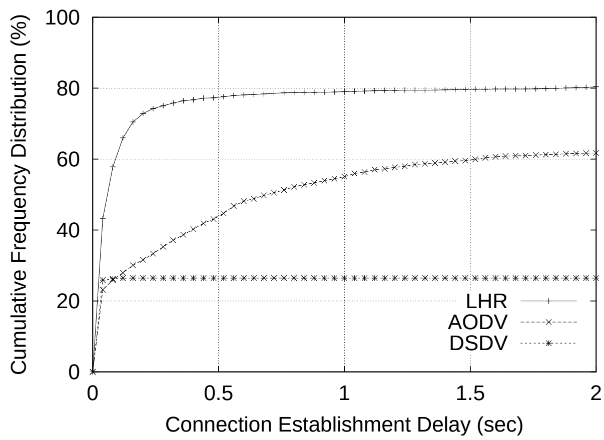
packet loss probability is 0% in the good state and 100% in the bad state. The unit time (T_G and T_B) of error and error-free states are defined as the time required for transmitting one data packet. In our simulations, this unit time was about 5.84 msec, which was calculated by dividing the data packet size (1460 bytes) by the transmission rate (2 Mbps). We employed the 1% error rate for all simulations, that was, the mean length of staying in the error-free state was set to 20,000 unit times ($=L_G$), and 200 unit times ($=L_B$) in the error state. For each of the simulation experiments, we calculated the state transition probabilities (p_{GG}, p_{GB}, p_{BB}, and p_{BG}) from the expressions derived in [20]. This model was employed to each link between nodes. The load of the network was defined as the total number of connections generated per second in the network. All connections were destined for one data collection terminal and the average of transmission data on each connection was 5 packets to simulate that the majority of connections are short-lived. The total simulation time was 210 seconds. All connection establishment requests arrived as a Poisson process and lasted for 100 seconds starting at 100 second in simulation.

Firstly, we investigated how many routes are maintained in one LHR node, that was described in Subsection 2.3. According to the results of simulations with 50 randomly placed stationary nodes, one LHR node maintained 1.9 shortest routes for the destination and 3.1 one-hop longer routes on average while a DSDV node always keeps 49 routes for all other nodes. We must note that this is a result when only one node is a destination of all other nodes. However, it is apparent that the way of DSDV wastes the network resource when we suppose a data-collecting application above an ad hoc network.

We measured TCP connection establishment delay, that is the time since TCP SYN generation until SYN+ACK receipt at a TCP source node, as a performance measure of routing protocols. Because of the space of paper, the result of data transmission time is not shown here. However, we note that trend of the result was the same as the result of connection establishment. 50 nodes are randomly distributed in the field in Figs. 5 to 7. Figures 5 and 6 show the cumulative frequency distribution of the number of connections that are established within the delay time indicated on the horizontal axis with each routing proto-

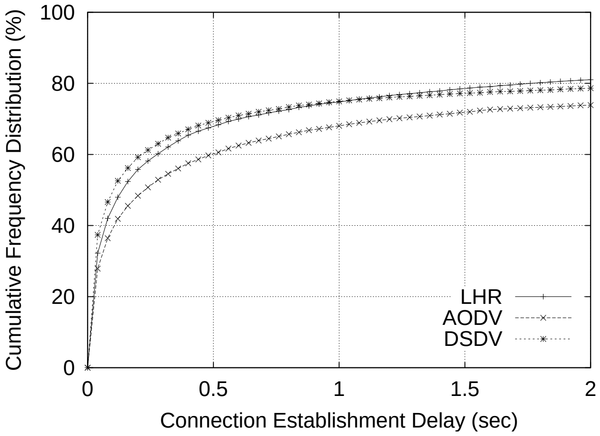


(a) Max Speed 0 m/sec

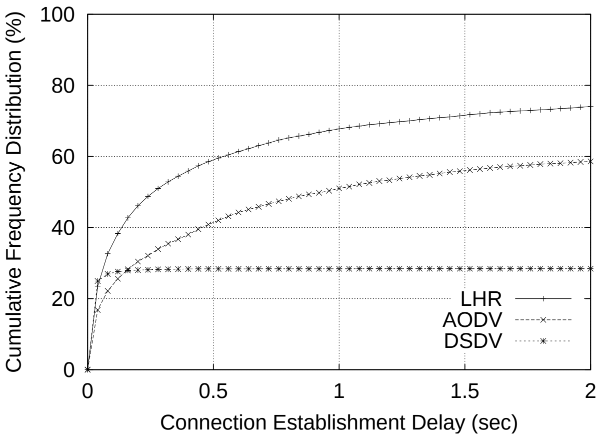


(b) Max Speed 20 m/sec

Fig. 5 Connection Establishment Delay (1 connections/sec)



(a) Max Speed 0 m/sec



(b) Max Speed 20 m/sec

Fig. 6 Connection Establishment Delay (5 connections/sec)

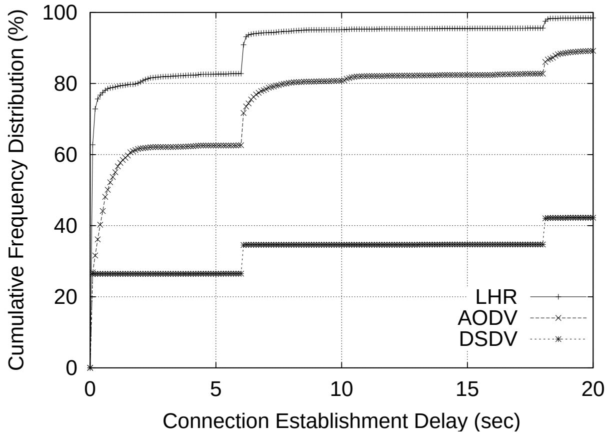
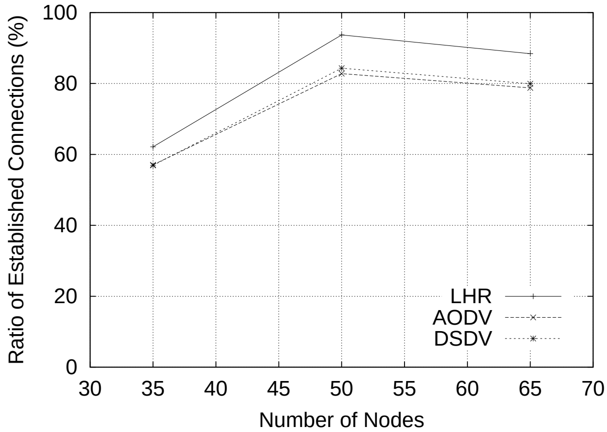


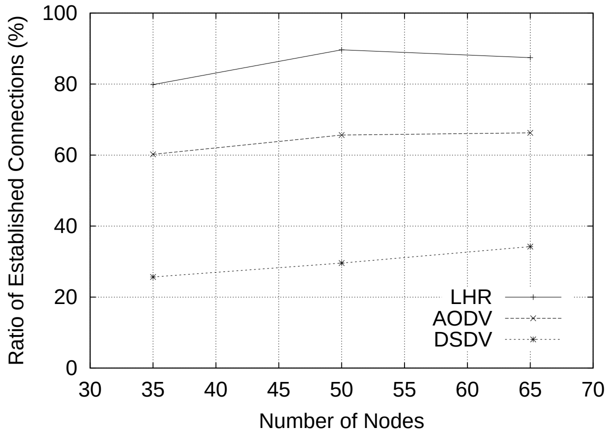
Fig. 7 Connection Establishment Retrials by SYN Retransmissions

col. Before we describe these figures, let us explain Fig. 7 to know what we see in these results. The higher limit of the x-axis was extended to 20 seconds in Fig. 7, and all simulation parameters are the same as in Fig. 5(b). We can see that the number of established connections increases sharply at 6 and 18 seconds. These were caused by retransmissions of TCP SYN packets, that is, the source nodes could not receive SYN+ACK packets until those time limits. Both on-demand protocols (LHR and AODV) can establish almost the same number of connections at the end of simulations, however, we must note that Fig. 7 indicates the connection establishment delay grows considerably if the routing protocol fails to transmit the first SYN and SYN+ACK packets. Based on the above result, we attended to the first handshake trial in Figs. 5 and 6. According to the simulation results shown in Figs. 5 and 6, LHR can establish more TCP connections in shorter time than other protocols. Firstly, we compare the performance of protocols which equip a table-driven route update method. As Figs. 5(a) and 6(a) show, LHR and DSDV can quickly establish more TCP connections in a stationary network. Since DSDV caches routes to all nodes in the network previously, it is a natural result that DSDV shows good performance in stable network. LHR shows good performance as well as DSDV, although LHR has no need to search routes previously. This is because route search packets of LHR can transmit TCP SYN and SYN+ACK packets simultaneously as described in Subsection 2.4. Seeing Fig. 5(b), the performance of DSDV which equips only a table-driven route update degrades in a high mobility network. This is because the periodic update cannot keep up with the changing network structure. Next, we compare the performance of protocols with an on-demand route search method. LHR outperforms AODV in all results because AODV generates too many route requests broadcasted by every source node and they waste the network resources. In such a situation, many nodes must wait packet transmissions long time to avoid the packet collisions. Furthermore, LHR shows much better performance in high mobility networks (Figs. 5(b) and 6(b)) because it starts route reconstruction from a node which detects a link disconnection. We can see the performance of LHR and DSDV degraded less than AODV in high load network in Fig. 6(b). This result means the routing packets of LHR and DSDV does not increase so much by the network load. According to these results, LHR is able to process more connections than other protocols in a given time period regardless of the network mobility and load.

When we think of constructing an ad hoc network consisting of many terminals, we also need to consider the node density in the network as an important network parameter. If the node density is low, the number of routes to the other nodes decreases. It causes no route to a destination, no substitute route against a link-disconnection, concentration of traffic to few routes, and increasing routes' length. In contrast, in high node density network, increase of routing traffic and interference from other nodes will degrade the network performance. Figure 8 shows the ratio of connections successfully established without TCP SYN retransmission versus the number of nodes distributed. Increase of node density has two important effects on the network performance and there is a trade-off between them. Firstly, increase of routing traffic degrades the network performance. See Fig. 8(a) as an example of this effect. If all nodes are stationary, the network performance degrades as the number of nodes increases from 50 to 65, since increasing routing messages block the data traffic. Secondly, seeing Fig. 8(b), increase of available routes gains the network capacity. In other words, large node density increases the number of routes and routes broken by node movement are more easily re-constructed. According to these results, we can see larger node density is available in higher mobility networks because route re-construction is easier if there are more routes to a destination.



(a) Max Speed 0 m/sec



(b) Max Speed 20 m/sec

Fig. 8 Node Density versus Successfully Established Connections without SYN Retransmission (3 connections/sec)

4. Conclusion

In this paper, we have proposed a new routing protocol, LHR, which was designed to decrease the routing latency for a network where many connections are short-lived. A sensor network is one promising application for which LHR is suitable. LHR adopts an on-demand route search and a proactive routing update. Packet receiving nodes are registered as active receivers, and only routes to them are exchanged. Against the link disconnections due to wireless error or node mobility, LHR maintains multiple routes for each destination to decrease the route re-search latency. In addition, to decrease initial connection establishment latency, LHR route request and route reply packet can carry the TCP connection establishment packets at one time.

In simulations, we have compared the connection establishment latency among LHR and other existing ad hoc routing protocols, and have shown that LHR has been capable of decreasing the latency of connection establishment and improving the performance for short-lived connections. Moreover, LHR has shown better performance in high-mobility and high-load network. It indicates that LHR has better performance as a mobile ad hoc network protocol.

References

[1] R.V. Boppana and S.P. Konduru, "An adaptive distance vector routing algorithm for mobile, ad hoc networks," Proc. IEEE INFOCOM 2001, March 2001.

[2] 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," Proc. ACM/IEEE MobiCom '98, pp.85–97, Oct. 1998.

[3] T. Goff, N. Abu-Ghazaleh, D.S. Phatak, and R. Kahvecioglu, "Preemptive routing in ad hoc networks," Proc. ACM SIGMOBILE 2001, pp.43–52, July 2001.

[4] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, "Scenario-based performance analysis of routing protocols for mobile ad-hoc networks," Proc. ACM/IEEE MOBICOM'99, Seattle, pp.195–206, Aug. 1999.

[5] N. Nikaein, H. Labiod, and C. Bonnet, "DDR – distributed dynamic routing algorithm for mobile ad hoc networks," Proc. ACM/IEEE MobiHoc 2000, Aug. 2000.

[6] M.R. Pearlman and Z.J. Haas, "Determining the optimal conguration for the zone routing protocol," IEEE J. Select. Areas Commun., vol.17, no.8, pp.1395–1414, Aug. 1999.

[7] A. Ahuja, S. Agarwal, J.P. Singh, and R. Shorey, "Performance of TCP over different routing protocols in mobile adhoc networks," Proc. IEEE VTC 2000-Spring, May 2000.

[8] T.D. Dyer and R.V. Boppana, "A comparison of TCP performance over three routing protocols for mobile ad hoc networks," Proc. ACM/IEEE MobiHoc 2001, Oct. 2001.

[9] G. Holland and N.H. Vaidya, "Analysis of TCP performance over mobile ad hoc networks," Proc. ACM/IEEE MOBICOM '99, Seattle, pp.219–230, Aug. 1999.

[10] 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, no.4, pp.249–271, Dec. 1999.

[11] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, "A feedback based scheme for improving TCP performance in ad-hoc wireless networks," Proc. IEEE ICDCS '98, Amsterdam, pp.472–479, May 1998.

[12] D. Kim, C.K. Toh, and Y. Choi, "TCP-BuS : Improving TCP performance in wireless ad hoc networks," Proc. IEEE ICC 2000, New Orleans, June 2000.

[13] C.E. Perkins, Ad Hoc Networking, Addison-Wesley, New York, Dec. 2000.

[14] C.E. Perkins, E.M. Belding-Royer, and S.R. Das, "Ad hoc on-demand distance vector (AODV) routing," IETF Internet Draft. `http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-10.txt`, Jan. 2002.

[15] B. Braden, "T/TCP – TCP extensions for transactions functional specifications," RFC 1644, July 1994.

[16] T. Yamamoto, M. Sugano, M. Murata, T. Hatauchi, and Y. Hosooka, "Performance improvement in ad hoc wireless networks with consideration to packet duplication," Proc. APCC 2001, pp.348–351, Sept. 2001.

[17] D.B. Johnson, D.A. Maltz, and Y.C. Hu, "The dynamic source routing protocol for mobile ad hoc networks (dsr)," IETF Internet Draft. `http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt`, July 2004.

[18] "The network simulator – ns-2." available at `http://www.isi.edu/nsnam/ns/`.

[19] E.N. Gilbert, "Capacity of a burst-noise channel," Bell System Technical Journal, vol.39, pp.1253–1265, Sept. 1960.

[20] J.R. Yee and E.J. Weldon, "Evaluation of the performance of error-correcting codes on a gilbert channel," IEEE Trans. Commun., vol.43, no.8, pp.2316–2323, Aug. 1995.

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. He is also now a research fellow of the Japan Society for the Promotion of Science. 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 Graduate School of Information Science and Technology, Osaka University in April 2004. He has more than three hundred papers of international and domestic journals and conferences. His research interests include computer communication networks, performance modelling and evaluation. He is a member of IEEE, ACM, The Internet Society, IEICE and IPSJ.