

Routing in Ad Hoc Networks for Processing Many Short-lived TCP Connections

Takayuki YAMAMOTO[†], Masashi SUGANO[‡], Masayuki MURATA^{††}

Abstract—Wireless ad hoc network is expected to be integrated with wired networks and many applications will communicate with TCP (Transmission Control Protocol). Many studies have been dedicated to improving the TCP performance 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 connections, the routing latency in ad hoc networks is considerable. We propose a new ad hoc routing protocol, the Low-latency Hybrid Routing protocol (LHR), that is suitable for a network where many TCP connections are short-lived. According to simulation results, LHR achieves a better performance than other existing routing protocols because LHR can establish TCP connections more quickly than other routing protocols.

I. INTRODUCTION

Ad hoc wireless networks are self-organized networks built with wireless terminals, which communicate with each other and exchange the network 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 analyze its characteristics and/or propose new routing methods (see, e.g., [1-5]). Some other studies have considered the use of TCP over ad hoc networks (e.g., [6-11]). 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 the steady-state throughput values. It is apparently inadequate because many TCP connections are short-lived. For example, it is reported in [12] that the average size of Web documents at several Web servers is about 10 [Kbytes]. Furthermore, in the sensor network, the amount of data on each connection is small, and major of the TCP connections would be short-lived. Since TCP is end-to-end communication protocol including wireless and wired terminals, its modification dedicated to the sensor network is not adequate for protocol migration. Instead, we should consider a new routing protocol in the ad hoc

network suitable for the short-lived TCP connections.

To improve the performance of short-lived connections, 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 TCP connection is persistent, the above problems do not affect the performance even in high-mobility and high traffic load environment. However, in actual TCP is never persistent, and most connections are short-lived.

A sensor network is one promising application over wireless ad hoc networks. In the sensor network, small amount of information is collected from many sensor terminals. Such a network model fits the most of characteristics of ad hoc networks, such as a distributed operation, scalability, and ease of maintenance. In the later Section III, we measure the packet loss probability of UDP and TCP to ascertain that the reliable transmission by TCP is important also in the network where the data traffic is intermittent from sensors. Many researchers have proposed various routing protocols in ad hoc networks. The Destination Sequenced Distance Vector (DSDV) [13] is a proactive protocol that each wireless node exchanges a route table periodically with neighbor nodes. The Ad hoc On-demand Distance Vector (AODV) [14] is a reactive (on-demand) protocol. When the route is unknown, the source node broadcasts a route query packet. The destination node sends a reply packet to the source node, and all intermediate nodes create a route entry to relay the reply packet. However, 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.

In this paper, we propose a new routing protocol that resolves the above problems and achieves low latency for the short-lived connections. We call it as the Low-latency Hybrid Routing (LHR) protocol that combines the on-demand route search and the proactive route maintenance. LHR adopts a quick route re-search method against a link disconnection. This is an advantage of LHR over the existing proactive and on-demand hybrid routing like ADV [1]. As a result LHR can process more TCP connections within a given time period. The remainder of this paper is organized as follows. We first describe the detail of LHR in Section II. Simulation setup and results are shown in Section III, and concluding remarks are made in Section IV.

[†] Department of Information Networking, Graduate School of Information Science and Technology, Osaka University, 1-3 Machikaneyama, Toyonaka, Osaka, Japan. E-mail: takymmt@ist.osaka-u.ac.jp . PHONE: +81-6-6850-6863 FAX: +81-6-6850-6868

[‡] Faculty of Comprehensive Rehabilitation, Osaka Prefecture College of Nursing, E-mail: sugano@osaka-hsu.ac.jp .

^{††} Cybermedia Center, Osaka University, E-mail: murata@anarg.jp

II. ROUTING PROTOCOL FOR SHORT-LIVED TCP CONNECTIONS

A. Decreasing the Overhead of Route Table Exchange

In some existing ad hoc routing protocols, more routes to terminals than actually used are maintained in the routing tables. 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 receivers and exchange them with neighbor nodes, so that the size of the route table 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.

B. Decreasing Latency for New Route Search

Proactive routing protocols can search a route promptly if it is cached in the route table [13]. However, it requires a long time to collect routing information from all over the network, and nodes cannot transmit packets to unknown destinations.

LHR uses an on-demand initial route search and a proactive route maintenance. 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. Then, they begin to multicast the HELLO messages periodically to neighboring nodes to update routes. All broadcasted RREP packets are given sequence numbers indicating the routes' freshness. 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.

C. Decreasing Latency for Route Re-search

The main originality of our proposed protocol is its route re-search method. 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 end-hosts. However,

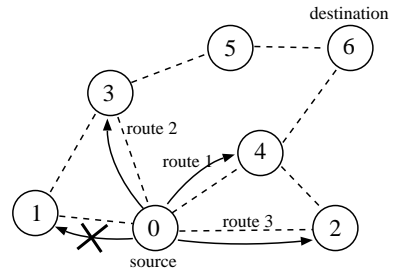


Fig. 1. Multiple routes' entry

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 failure. 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. In short, 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. We describe these methods in more detail below.

In an initial route search, described in Subsection II-B, a node may receive the identical RREP from two or more neighbor nodes. It indicates that there are multiple routes to the destination. The node caches these routes to recover a link disconnection quickly. In more detail, 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 re-activated when a packet is received through the link again. 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.

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 Figure 1 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 [15]. Based on the result in [15], 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. If all transmission

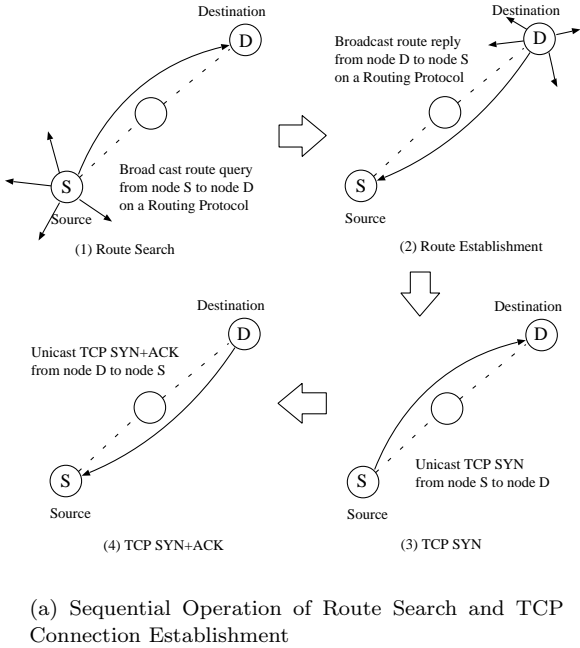


Fig. 2. Connection Establishment Flow

trials fail at last, node 0 initiates a RREQ packet to search a fresh route.

D. 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 connection time, the time for connection establishment becomes considerable especially in short-lived TCP connections.

In LHR, two message packets are broadcasted at TCP end-hosts when the route to destination is unknown. Therefore, at the beginning of TCP connection on a new route, they must exchange four packets (two routing packets and two TCP packets) as before the source node receives the SYN+ACK packet. It results a considerable latency for

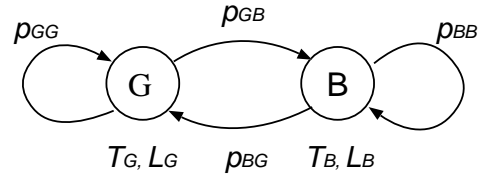


Fig. 3. Two-state Transition Error Model

short-lived connections. We can decrease it by integrating TCP connection establishment with route search in LHR. See Figure 2. When the node initiating the SYN packet finds no available route, it broadcasts the RREQ packet carrying the SYN packet together. The RREP packet also carries the SYN+ACK packet. Thus, the connection establishment time can be decreased. It is inevitable that the network load increases. However, this is acceptable because we now aim at decreasing the latency for short-lived connections at the expense of increased traffic load.

III. SIMULATION EXPERIMENTS

We implemented LHR using an ns-2 network simulator [16]. 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. We simulated a 500 x 2000 m² network field that consisted of randomly placed 50 nodes. Their mobility pattern was based on a *random way-point* model [2]. Wireless error was modeled by a two-state transition error model (Figure 3) known as the Gilbert model [17]. For each of the simulation experiments, we calculated the error rate from the expressions derived in [18]. The 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 was about 5.84 msec. We employed the 1% error rate for all simulations, that is, the mean length of staying in the error-free state was set to 20,000 slots, and 200 slots in the error state. The load of the network was defined as the total number of connections generated per second in the network. All connections are destined for one data collection terminal and the average of transmission data on each connection is 5 packets.

We first compared the packet loss probability of UDP and TCP in the simulated network. In these simulations, we saw the importance of reliable transmission by TCP in a low-load network like a sensor network. Tables I and II show the results of the network where 1 and 5 connections are generated per second respectively. Using UDP, the packet loss is considerably high also in the low-mobility and low-load network. TCP achieves low packet loss rate with any type of routing protocols although TCP increases the total amount of network traffic by connection setup and acknowledging data transfer. This is important for collecting time-dependent sensing data.

Routing	LHR				AODV				DSDV			
Mobility (m/s)	0		20		0		20		0		20	
Transport	TCP	UDP	TCP	UDP	TCP	UDP	TCP	UDP	TCP	UDP	TCP	UDP
Packet Loss (%)	0	3.8	0	11	0	9.7	0.3	16.7	0	6.2	0.3	63.9

TABLE I
COMPARISON OF UDP AND TCP FOR THE INTERMITTENT TRAFFIC (1 CONNECTION/SEC)

Routing	LHR				AODV				DSDV			
Mobility (m/s)	0		20		0		20		0		20	
Transport	TCP	UDP	TCP	UDP	TCP	UDP	TCP	UDP	TCP	UDP	TCP	UDP
Packet Loss (%)	1.4	7.2	0.9	12.4	2.5	17.3	1.3	17.8	1.1	10.4	0.3	63.7

TABLE II
COMPARISON OF UDP AND TCP FOR THE INTERMITTENT TRAFFIC (5 CONNECTIONS/SEC)

Next, we measured the TCP connection establishment delay, that is the time since TCP SYN generation until SYN+ACK receipt at the TCP source node, with LHR, AODV, and DSDV routing protocols respectively. Figures 4 and 5 show the cumulative frequency distribution of the number of connections that could be established within the latency indicated on the horizontal axis. According to the simulation results, LHR is capable to connect more TCP connections in shorter time than other protocols. As Figures 4(a) and 5(a) show, routing protocols with table-driven route updates (LHR and DSDV) can quickly establish many TCP connections in a network with stationary nodes. However, if the number of link disconnections grows as a result of high node mobility, DSDV cannot search a correct route quickly ((b) and (c) of Figures 4 and 5), since proactive protocols are less able to accommodate the frequently changing network topology. On-demand routing protocols are adequate for such high-mobility network. LHR outperforms AODV also in such a network, because AODV generates too many route requests broadcasted by every source node and they waste the network resources. Thus, LHR can process more connections than other protocols in a given time period.

When we think of constructing an ad hoc network, 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 other nodes decreases. It causes no route to a destination, no substitute route against a link-disconnection, concentration of traffic to few routes, and increase the hop counts of routes. In contrast in high node density network, increase of routing traffic and interference from other nodes will degrade the network performance. Figure 6 shows how many connections are successfully established without TCP SYN retransmissions. We changed the number of nodes placed in the same field for these simulations. 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. In high-mobility network, the result indicates that the larger node density than in stationary network is acceptable. This is because, when the node density becomes large, the probability that a route broken by node movement is re-constructed increases, while the routing traffic increases.

IV. CONCLUSION

In this paper, we have proposed a new routing protocol that is applicable to the existing Web systems where many TCP connections are short-lived. The sensor network is another important application where the TCP connections for collecting a small amount of data from many sensor terminals are major within the network. In those networks, we need to decrease the connection and transmission latency for the short-lived connections.

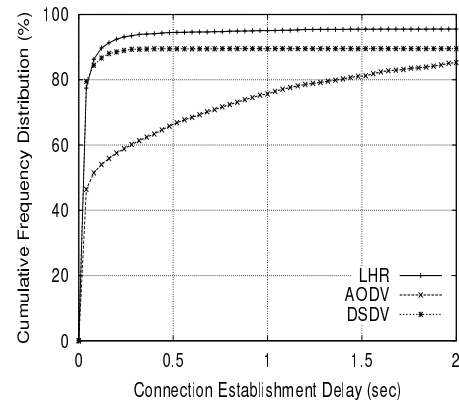
Our protocol LHR adopts an on-demand route search and a proactive routing update. Packet receiving nodes are registered as active receivers, and only routes to them are exchanged. Against the link disconnections due to wireless error or node mobility, LHR maintains multiple routes for each destination 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 the same time. These features are capable of decreasing the latency of connection establishment and improving the performance for short-lived TCP connections.

ACKNOWLEDGEMENTS

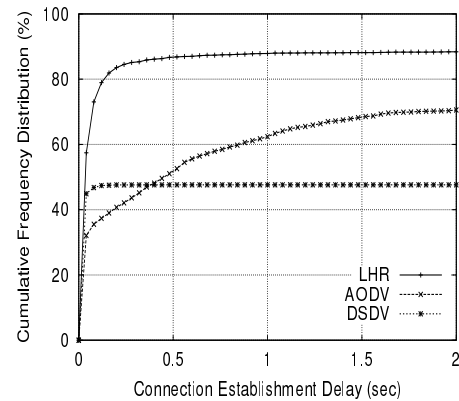
This work was partly supported by Special Coordination Funds for promoting Science and Technology of the Ministry of Education, Culture, Sports, Science and Technology of Japan.

REFERENCES

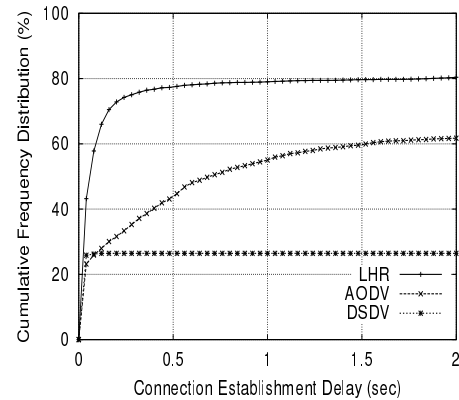
- [1] R. V. Boppana and S. P. Konduru, "An adaptive distance vector routing algorithm for mobile, ad hoc networks," in *Proceedings of the INFOCOM 2001*, Mar. 2001.
- [2] J. Broch, D. A. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," in *Proceedings of MOBICOM '98*, pp. 85–97, Oct. 1998.
- [3] 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 the MOBICOM 99*, (Seattle), pp. 195–206, Aug. 1999.
- [4] N. Nikaiein, H. Labiod, and C. Bonnet, "DDR – Distributed Dynamic Routing Algorithm for Mobile Ad Hoc Networks," in *Proceedings of the MobiHOC 2000*, 2000.
- [5] M. R. Pearlman and Z. J. Haas, "Determining the Optimal Conguration for the Zone Routing Protocol," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1395–1414, Aug. 1999.
- [6] 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 the VTC 2000 spring*, May 2000.
- [7] 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 the ICDCS '98*, pp. 472–479, May 1998.
- [8] D. Kim, C.-K. Toh, and Y. Choi, "TCP-BuS : Improving TCP Performance in Wireless Ad Hoc Networks," in *Proceedings of the ICC 2000*, June 2000.
- [9] T. D. Dyer and R. V. Boppana, "A Comparison of TCP Performance over Three Routing Protocols for Mobile Ad Hoc Networks," in *Proceedings of the Mobihoc 2001*, Oct. 2001.
- [10] T. Goff, N. Abu-Ghazaleh, D. S. Phatak, and R. Kahvecioglu, "Preemptive routing in ad hoc networks," in *Proceedings of the MOBICOM 2001*, pp. 43–52, 2001.
- [11] G. Holland and N. H. Vaidya, "Analysis of TCP Performance over Mobile Ad Hoc Networks," in *Proceedings of the ACM/IEEE MOBICOM'99*, pp. 219–230, Aug. 1999.
- [12] 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, Dec. 1999.
- [13] C. E. Perkins, *AD HOC NETWORKING*. Addison-Wesley, 2001.
- [14] C. E. Perkins, E. M. Belding-Royer, and S. R. Das, "Ad hoc on-demand distance vector (aodv) routing," in *IETF Internet Draft*. <http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-10.txt>, Jan. 2002.
- [15] T. Yamamoto and M. Sugano and M. Murata and T. Hatauchi and Y. Hosooka, "Performance improvement in ad hoc wireless networks with consideration to packet duplication," in *Proceedings of the APCC 2001*, pp. 348–351, Sept. 2001.
- [16] "The Network Simulator - ns-2." available at <http://www.isi.edu/nsnam/ns/>.
- [17] E. N. Gilbert, "Capacity of a Burst-Noise Channel," *Bell System Technical Journal*, vol. 39, pp. 1253–1265, Sept. 1960.
- [18] 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, Aug. 1995.



(a) Max Speed 0 m/sec

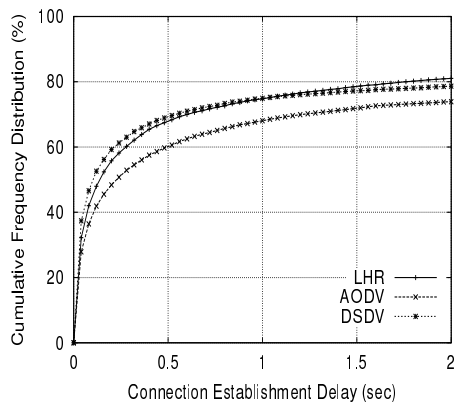


(b) Max Speed 5 m/sec

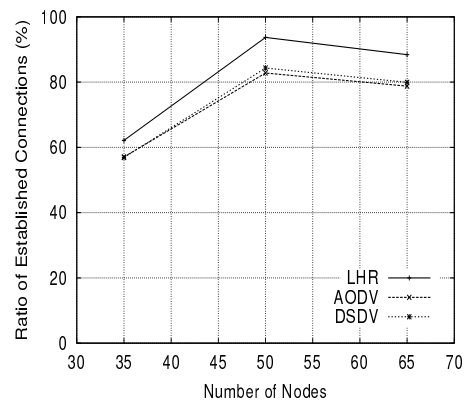


(c) Max Speed 20 m/sec

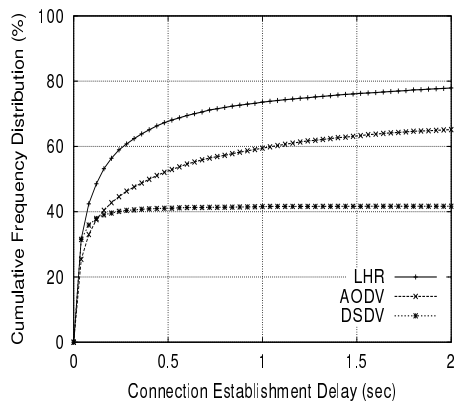
Fig. 4. Connection Establishment Delay (1 connections/sec)



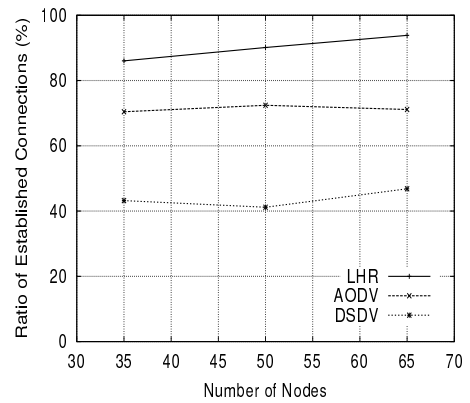
(a) Max Speed 0 m/sec



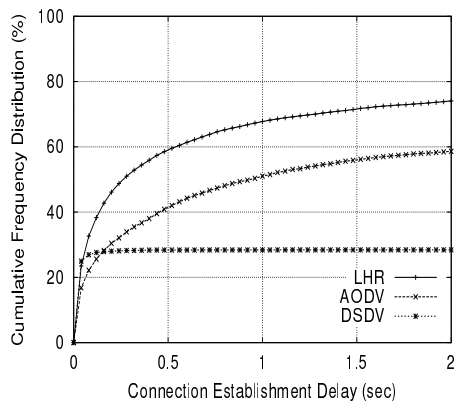
(a) Max Speed 0 m/sec



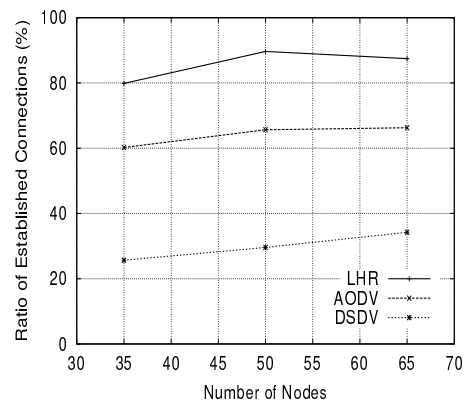
(b) Max Speed 5 m/sec



(b) Max Speed 5 m/sec



(c) Max Speed 20 m/sec



(c) Max Speed 20 m/sec

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

Fig. 6. Node Density versus Successfully Established Connections without SYN Retransmission (3 connections/sec)