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

Takayuki YAMAMOTO, Masashi SUGANO, Masayuki MURATA

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]). Some other studies have considered the use of TCP over ad hoc networks (e.g., [2, 3]). 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 [4] 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.

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 [5]. As a result LHR can process more TCP connections within a given time period.

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 [6] 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 active receivers like ADV [5]. Only routes to active receivers are maintained and exchanged with neighbor nodes, so that the size of the route table can be much decreased. While an initial connection to an inactive receiver takes long latency with the table driven routing protocols, LHR adopts another on-demand routing mechanism to decrease the connection latency to the inactive receiver.

T. Yamamoto is with 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

B. Decreasing Latency for New Route Search

Proactive routing protocols can search the route promptly if available [6]. However, they need large time to collect routing information all over the network, and nodes cannot transmit packets to unknown destinations.

In LHR, on-demand initinal route search and proactive route maintenance are used. 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 route's 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, 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.

In a initial route search described in Subsection II-B, a node may receive the RREP packet from two or more

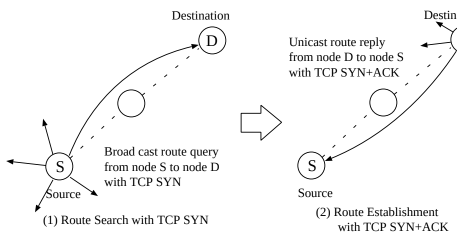


Fig. 1. Connection Establishment Flow

neighbors, it indicates that there are multiple routes to the destination. The node caches those routes to recover a link disconnection quickly. If the packet transmission through any cached route cannot succeed, the node initiates a RREQ packet to find an available route to the destination.

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 comes a considerable latency for short-lived connections. We can decrease it by integrating TCP connection establishment with route search in LHR. See Figure 1. 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.

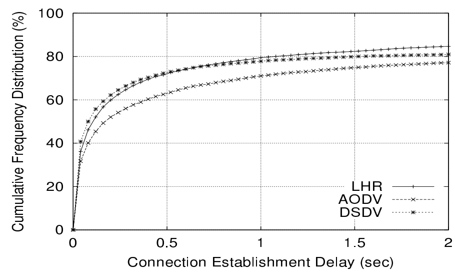
III. Simulation Experiments

We implemented LHR using an ns-2 network simulator [7]. We used AODV and DSDV implementations of ns-2. An IEEE 802.11 was employed at the link- and physical-layer. We simulated a 500 x 2000 m network field that consisted of randomly placed nodes. Their mobility pattern was based on a *random way-point* model [8].

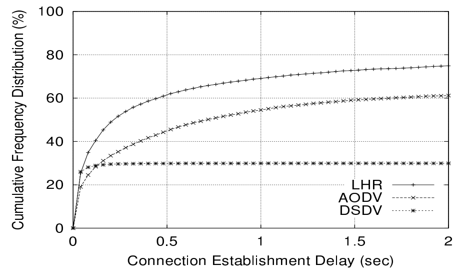
We measured the TCP connection establishment delay for performance comparison. This delay is the time from TCP SYN generation to SYN+ACK receipt at the TCP source node. All TCP connections are destined for one data collection terminal and their average amount of transmission data is 5 packets. Figure 2 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. This result indicates that more connections can be processed by LHR than others in a given time period.

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



(a) Max speed 0 m/sec



(b) Max speed 20 m/sec

Fig. 2. Connection Establishment Delay (5 connections/sec)

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 a proactive routing update. Packet receiving nodes are registered as active receivers, and only routes to them are exchanged. Routes to inactive receivers are established on demand basis. For the link disconnection 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.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

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

[7] "The Network Simulator - ns-2." available at `http://www.isi.edu/nsnam/ns/`.

[8] 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.