Low-latency Routing for Short-Lived TCP Connections in Wireless Ad Hoc Networks

Takayuki YAMAMOTO, Masashi SUGANO, Masayuki MURATA

I. Introduction

A wireless ad hoc network does not require wired infrastructure or network management terminal. That is, it is self-organized only with wireless terminals that exchange network information, maintain routes, and construct a multi-hop distributed network. In the wireless ad hoc network, there is no need for wired infrastructure or network management terminal. Therefore, quickly and easily we can build a large-scale network with flexibility in adding and removing terminals. Such a wireless infrastructure is applicable to, e.g., distributed computing, disaster recovery, and military operation. A sensor network system, which collects information from many terminals, is another good example of the ad hoc network.

In the past, many studies have been dedicated to analyze characteristics or to propose new routing methods of the wireless ad hoc networks. See, e.g., [1-3]. With expectation that it is integrated with wired networks by using TCP, the performance of TCP over ad hoc networks has been studied in [4-8]. However, most of those studies 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 [9] that the average size of Web documents at several Web servers is about 10 [KBytes]. Especially, 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 TCP is persistent as in the existing routing protocols, 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.

In this paper, we propose a new routing protocol resolving the above problems and establishing the low latency for short-lived connections. We call it as Low-latency Hybrid Routing (LHR) protocol that combines on-demand route search and proactive route maintenance. LHR also has a capability of decreasing latency for another route search in case of link disconnection, which will be explained in more detail in the next section.

II. Routing Protocol for Sensor Networks

A. Decreasing the Size of Route Table Exchange

In most of existing routing protocols in ad hoc networks, nodes that collect data relating to the routing information are pre-specified, and those are not changed during their activities. One example is DSDV [10] that maintains routes to all nodes. However, it 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 [11]. Only routes to active receivers are maintained and exchanged with neighbor nodes, so that the size of the route table can be much decreased. An initial connection to an inactive receiver takes long latency in this algorithm because it needs a broadcast route search. However, LHR adopts another mechanism to decrease the connection latency to the inactive receiver. We describe it in Subsection II-D.

B. Decreasing Latency for New Route Search

On-demand routing protocols begin route search on packet transmission demand, which tends to require large latency. On the contrary, proactive routing protocols can search the route earlier if available [10]. However, they need large time to collect routing information all over the network, and nodes cannot transmit packets to unknown destinations. ADV avoids this problem by maintaining a route table exchange to active receivers and initiating broadcast route search to inactive receivers.

LHR adopts the similar method for route maintenance and discovery. In LHR, a source node broadcasts Route-Request (RREQ) packet when it has a data packet to an inactive receiver. The target destination receiving this packet broadcasts Route-Reply (RREP) packet. All nodes receiving these two packets register the target destination node as an active receiver. Then, they begin to broadcast HELLO messages periodically to neighboring nodes, and updates routes proactively. All broadcasted RREP packets are given sequence numbers indicating the route's freshness. However, in case of the link failure, nodes must wait for the neighbors' route update message. To decrease

T. Yamamoto is with Department of Informatics and Mathematical Science, Graduate School of Engineering Science, Osaka University, 1-3 Machikaneyama, Toyonaka, Osaka, Japan. E-mail: takymmt@ics.es.osaka-u.ac.jp . PHONE: +81-6-6850-6616 FAX: +81-6-6850-6589

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 as listed below for recovering routes against the link disconnection, which is caused by node movement and/or changes in the wireless environment.

1. Exchanging the routing table with neighbor nodes. Another available route can be found from the routing table. Its problem is that it has no way to search another route on demand.
2. A route error message is initiated to the source node. On receiving this message, the source node begins another route request process. On-demand routing protocols such as DSR [12] adopt it. 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 link failure. Though this method sometimes makes 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. This gives another route which will be available quickly.

We adopt method 3 and 4 together. Nodes suffering a link disconnection first try retransmission through method 4. If no more routes are available, they try method 3. In what follows, we will describe these methods in more detail.

As we have described in Subsection II-B, the node receiving a RREQ packet broadcasts a RREP packet. When a node receives the PREP packet from multiple neighbors, it indicates that there are multiple routes to the destination. Then, they cache these routes for the link disconnection. Nodes can also get multiple routes in routing table exchange with neighbor nodes.

When a node receives route information by the RREP packet or the HELLO message 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 II-B, with those of the route entries it maintains. If the former one has a bigger sequence number, the node deletes all old route entries to the destination node and begins to use new route. If the former one has the same sequence number as the latter ones, the node can use that for a backup route. If it has the smaller sequence number, the node simply ignores it.

With this multiple route maintenance mechanism, the number of route entries 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 S to node D has two hops. The node S 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 systems [13]. Based on the result in [13], the shorter route is given a higher priority. When node S has a packet destined for node D, node S first tries the transmission on route 1, the shortest route to node D. If it fails, node S removes the route 1, and tries the second shortest route. If all transmission trials fail at last, node S initiates a RREQ packet.

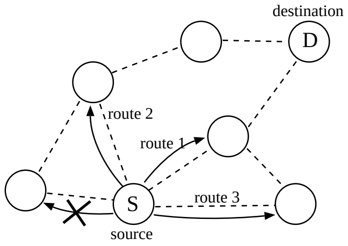


Fig. 1. Multiple routes' entry

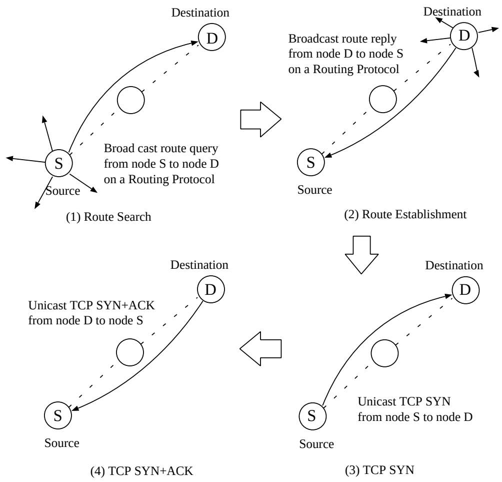
D. TCP Connection Establishment Integrated with Routing Protocol

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.

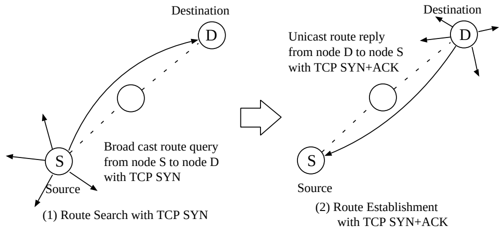
In LHR, two message packets are broadcasted at TCP end-hosts when the route to destination is unknown. Therefore, at the beginning of TCP connection on a new route, they must exchange four packets as illustrated in Figure 2(a) before the source node receives SYN+ACK. It comes a considerable latency for short-lived connections. We can decrease it by integrating TCP connection establishment with route search in LHR. See Figure 2(b). 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. Conclusion

In this paper, we have proposed a new routing protocol that can adapt to the actual network, by which we mean that it is applicable to the existing Web systems in which many TCP connections are short-lived. The sensor network is another important application where TCP connections collecting a small amount of data from terminals are major within the network. In those networks, it is important to



(a) sequential operation of route search and TCP connection establishment



(b) route search packets with SYN and SYN+ACK packets

Fig. 2. Connection establishment flow

decrease the connection and transmission latency for short-lived connections.

Our protocol LHR adopts proactive routing updates. 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.

We are now conducting the simulation experiment to identify the performance of our proposed routing protocol, which will be included in the final paper.

References

[1] 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.

[2] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T. Chen, "Scalable Routing Strategies for Ad Hoc Wireless Networks," *IEEE J. Select. Areas Commun.*, pp. 1369–1379, Aug. 1999.

[3] M. R. Pearlman and Z. J. Haas, "Determining the Optimal Configuration for the Zone Routing Protocol," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1395–1414, Aug. 1999.

[4] 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.

[5] 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.

[6] 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.

[7] G. Holland and N. H. Vaidya, "Analysis of TCP Performance over Mobile Ad Hoc Networks," in *Proceedings of the MOBICOM 99*, (Seattle), pp. 219–230, 1999.

[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] Masahiko Nabe, Masayuki Murata, and Hideo 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.

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

[11] 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.

[12] D. B. Johnson, D. A. Maltz, Y.-C. Hu, and J. G. Jetcheva, "The dynamic source routing protocol for mobile ad hoc networks," in *IETF Internet Draft.* `http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-05.txt`, 2001.

[13] 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, Sep. 2001.