A Scalable Routing Protocol for Ad Hoc Networks

Huaizhi Li
Department of Computer Science
University of Kentucky
Lexington, KY 40506
Email: hli3@cs.uky.edu

Mukesh Singhal
Department of Computer Science
University of Kentucky
Lexington, KY 40506
Email: singhal@cs.uky.edu

Abstract—Ad hoc networks, which do not rely on any infrastructure such as access points or base station, can be deployed rapidly and inexpensively even in situations with geographical or time constraints. So ad hoc networks have attractive applications in both military and disaster situations and also in commercial uses like sensor networks or conferencing. In ad hoc networks, each node acts both as a router and as a host. The topology of an ad hoc network may change dynamically, which makes it difficult to design an efficient routing protocol. Nowadays, more and more wireless devices are used, which can form large ad hoc networks. It is important to design scalable routing protocol for ad hoc networks. In this paper we present Anchor Based Routing Protocol (ABRP), a scalable routing protocol for ad hoc networks. It is a hybrid routing protocol, which combines table based routing strategy with geographic routing strategy. Simulation result shows that it is efficient and scale well to large network.

I. INTRODUCTION

An ad hoc network is a set of wireless mobile nodes that form a dynamic autonomous network without the intervention of centralized access points or base stations. Unlike traditional wireless networks, ad hoc networks require no fixed network infrastructure and can be deployed as multi-hop packet networks rapidly and with relatively low expense. Thus such networks can be very useful in scenarios where natural condition or time restrains makes it impossible to have infrastructure pre-deployed. Examples of applications could be battlefields, emergency services, conference rooms, or even home and office devices.

Mobile nodes in an ad hoc network have limited radio transmission range. Nodes that are unable to communicate directly with each other require intermediate nodes to forward packets for them. Each node acts both as a router and as a host. The function of a routing protocol in ad hoc network is to establish routes between different nodes. Much research work has been done on routing in ad hoc network. Here we classify the existing ad hoc routing protocols as table based routing protocol and non-table based routing protocol. The table based routing protocols can be categorized as proactive routing protocol, reactive routing protocol and hybrid routing protocol. Proactive routing protocol, like DSDV [1] and TBRPF [2], periodically exchange routing information in the whole network so that routes between different nodes are dynamically maintained. Therefore proactive routing protocols have low latency and the routes are reliable. Yet they have high overhead since even if there is no data traffic in the network routing messages are still kept exchanging. Thus these protocols can not scale well with increasing of network size. Reactive routing protocols, such as AODV [3], DSR [4] and TORA [5], exchange routing information only to establish routes between nodes that have data traffic. So the overhead is lower than proactive routing protocols. Yet in large network and high mobility situation, path is long and link may frequently break, so overhead may be high. Hybrid routing protocol, like ZRP [6] is combination of proactive and reactive routing approaches.

In contrast to table based routing protocols, geographic routing protocols [7]–[10] utilize location information to route packets. They do not exchange routing message to establish routes. We classify these routing protocols as non-table based routing protocols. When a node has data packets to be transmitted to another node, forwarding desion is made based on the position of the destination and the position of the neighbors of the forwarding node. Thus position based routing protocol is stateless routing protocol. It is scalable and has low overhead. But to determine the physical position, each node must be equipped with GPS [11] or use other type of positioning service [12]–[14].

With the rapid development of wireless technology, more and more wireless devices enter people’s daily life. This increases the need of deployment of ad hoc network with large number of node. Thus the scalability of an ad hoc routing protocol becomes an important issue. Lee et al. [15] studied the scalability of AODV. Cluster based routing protocols, such as [16] attempt to increase the scalability of routing protocols in ad hoc networks. Yet the overhead to maintain the cluster structure is still too high.

In this paper, we present the Anchor Based Routing Protocol (ABRP), a scalable routing protocol for ad hoc networks. ABRP combines table based routing strategy with geographic routing strategy. Simulation result shows that comparing with AODV, ABRP scales well to large network, has low overhead and resonable end-to-end delay.

The rest of the paper is organized as follows. In Section II, we present our Anchor Based Routing Protocol. Section III describes the simulation environment and presents the simulation results. Related work is presented in Section IV. Section V offers concluding remarks and points out future directions.
II. THE ROUTING PROTOCOL

In this section we describe our routing protocol. The protocol consists of three parts: a location based clustering protocol, an intra-cell routing protocol, and an inter-cell routing protocol, which will be introduced respectively. Each node in the network maintains two kinds of routing tables: inter-cell routing table and intra-cell routing table.

A. Location Based Clustering Protocol

The purpose of this clustering protocol is to divide the network region into many cells. In our protocol, we assume that several physical locations in the network are known. For example, in a campus region, the location of some buildings is known. Each location is called an anchor. An anchor becomes center of a cell. Each anchor has coordinates. We also assume that there is always a node close to each location, e.g., within a certain range, like 20 meters. This is a reasonable assumption, since any computer in a building can have this role. This node is called agent of that anchor. Each anchor is assigned an ID, like, 1, 2, and 3. Each node in the network knows the ID and coordinates of all the anchors. It can be thought as each node having a map of anchors. The agent of each anchor periodically broadcasts an ANCHOR-ANNOUNCEMENT message to announce the ID of its anchor. The TTL (time to live) of the ANCHOR-ANNOUNCEMENT message is 1 or 2 hops. When a node receives the ANCHOR-ANNOUNCEMENT message, it joins that cell and uses that anchor’s ID as its cell ID. It is possible that a node may receive two ANCHOR-ANNOUNCEMENT messages from two different agents. In this case, it picks a smaller ID. With this approach, the network is naturally divided into different cells. The radius of each cell is 1 or 2 hops. After a node obtains a cell ID, it also knows the neighbor cells of its own cell, since it knows the location of all the anchors. To ensure that all the cells can cover the whole network, i.e., there are no blind zones in the network, a sufficient number of anchors should be used.

B. Inter-cell Routing Protocol

The purpose of inter-cell routing protocol is to let each node dynamically maintain multiple paths to its neighbor cells. The protocol has a flavor of proactive routing. The inter-cell routing table entries are tuples in the form of (cell ID, bridge node ID, sequence number, next hop, distance).

The Location Based Clustering Protocol ensures that each node has a cell ID. Each node also knows its neighbor cells. To discover paths to its neighbor cells, a node broadcasts a route request message, INTER-RREQ: {source_addr, source_ID, seqn, ID_list}, in which source_addr is the address of the source node; source_ID is the cell ID of the source node; seqn is sequence number; and ID_list is a list of ID of neighbor cells. When a node receives the INTER-RREQ, it first looks at the message to see if it is a new request from that source node. If not, the node silently discards the route request. For a new route request, the node first creates a reverse route entry for the source node and adds the entry to its intra-cell routing table. Then the node looks at the ID_list field of the route request. If its ID is in the list, it becomes a bridge node to this neighbor cell for the source node and generates a route reply message, INTER_RREP: {bridge_addr, cell_ID, seqn}, in which bridge_addr is the address of this node; cell_ID is the ID of this neighbor cell which is equal to the bridge node’s cell ID; seqn is sequence number to prevent looping. The INTER_RREP is sent back to the source node. If a node’s cell ID is the same as the source_ID, the node is an intermediate node. It looks at its routing table to see if it has valid paths to the neighbor cells listed in the INTER_RREQ. If it does, it generates an INTER_RREP and unicasts the reply back to the source node along the reverse path. An intermediate node may have paths to different neighbor cells, so the reply generated by an intermediate node contains a list of valid paths, each of which is described by (bridge_addr, cell_ID, seqn). When an intermediate node receives the INTER_RREP, it creates forward route entries for the neighbor cells contained in the route reply and adds the route entries to its inter-cell routing table. If an intermediate node has valid paths to all the neighbor cells in the ID_list of the INTER_RREQ message, it discards the INTER_RREQ, since all the requested routes have been found. Otherwise it rebroadcasts the INTER_RREQ. Before rebroadcasting, it trims the ID_list of the INTER_RREQ: removes the cell IDs that it has paths to. So the rebroadcast INTER_RREQ only contains the cell IDs that the intermediate node does not have paths to. With this approach, we can reduce overhead. Eventually the source can receive the INTER_RREP. And routes to neighbor cells have been established. It is likely that an INTER_RREQ reaches several different nodes in a neighbor cell. All these nodes become bridge nodes for the source node. A node sends out an INTER_RREQ to a particular neighbor cell only when all the paths to that neighbor cell are invalid. It can be seen that an INTER_RREQ can only reach one hop beyond the boundary of the cell of the source node. So our inter-cell routing protocol has low overhead.

C. Intra-cell Routing Protocol

Intra-cell routing protocol is for routing within the same cell. It is an on demand routing protocol. The intra-cell routing table entries are tuples in the form of (destination, sequence number, next hop, distance).

When a node has a data packet destined to another node within the same cell and it does not have a valid route, it generates an intra-cell route request message, INTRA_RREQ: {source, source_ID, destination, sequence number}, and broadcasts it. When a node receives the INTRA_RREQ, it first checks if its cell ID is same as the source_ID in the INTRA_RREQ. If not, it implies that the INTRA_RREQ is from other cell (Nodes at the boundary of a cell may receive INTRA_RREQ from neighbor cell). It simply discards the INTRA_RREQ. Otherwise, it creates a reverse route entry for the source node and adds it to its intra-cell routing table. Then it checks if it is the required destination node. If so, it generates an intra-cell route reply, INTRA_RREP: {destination, source, sequence number} and...
unicasts it to the source node. If it is not the destination node, it may have a valid route to the destination. In this case it also generates an INTRA_RREP packet and unicasts the reply to the source. If it does not have a valid route to the destination, it rebroadcasts the INTRA_REQ.

When an intermediate node receives the INTRA_RREP, it generates a forward route entry for the destination node and adds the entry to its intra-cell routing table. It then forwards the INTRA_RREP to the source. After the source node receives the INTRA_RREP, it can use the route to send data packet. It can be seen that the propagation of intra-cell routing messages is confined within each cell.

Note that our inter-cell routing protocol (section II-B) not only discovers multiple routes to the neighbor cells of a source node, it also naturally lets other nodes that receive the INTER_RREQ establish a intra-cell route to the source node.

D. Data Packet Routing

We now explain how a data packet is transmitted from a source node to a destination node. We assume that a source node knows the cell ID of the destination node. This can be done by a location service, which is a necessary component in geographic routing protocols. Our protocol’s requirement is easier to meet since we only need the destination’s cell ID, not its exact location. When the source node has a data packet to be sent, it first checks the destination’s cell ID to see if they are in the same cell. If so, the source node utilizes intra-cell routing protocol to transmit the packet. If not, the source node calculates the distance between the anchors of its neighbor cells and the anchor of the destination node. The neighbor cell that has the shortest distance is the next cell that the data packet will be forwarded to. The inter-cell routing protocol has established a route or multiple routes to the next cell. The source node picks the shortest one, adds the destination’s cell ID in the header of the data packet and forwards the packet to its bridge node in the next cell. When the bridge node receives the data packet, it processes the data packet with the same procedure, i.e., comparing its cell ID with the destination’s cell ID and using either the intra-cell routing protocol or the inter-cell routing protocol to forward the data packet. Finally, the data packet reaches a bridge node of the destination’s cell. The bridge node delivers the packet to the destination using the intra-cell routing protocol.

If the source and the destination are in different cells, the path between them consists of two kinds of nodes: (1) bridge nodes, (2) intermediate nodes between two bridge nodes. At each intermediate cell between the source and the destination, the bridge node picks the next cell to forward the packet. It is possible that the bridge node does not have a valid route to the neighbor cell with the shortest distance; it can use the neighbor cell with the second shortest distance as the next cell. At an intermediate node, if the route picked by the bridge node is down, the intermediate node can use other valid routes to the same next cell. These measures improve the robustness of our routing protocol.

### III. PERFORMANCE ANALYSIS

#### A. Simulation Environment

We evaluated the performance of ABRP by comparing ABRP with Ad-hoc On-Demand Distance Vector (AODV) Routing protocol [3]. We implemented ABRP using ns2 simulator [17] and the wireless extensions developed by CMU. In the experiments, the MAC layer is the IEEE 802.11 MAC protocol with Distributed Coordination Function (DCF) [18], which uses Request-to-send (RTS) and Clear-to-send (CTS) control frames for unicast packet. The Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) is the access frame. The characteristics of radio model is similar to Lucent’s WaveLAN [19], which is a shared-media radio with 2Mbit/sec transmission rate and a normal radio range of 250 meters.

#### B. Traffic and Mobility Patterns

In this simulation, we used two field configurations as shown in Table I. We varied the field size to keep the node density approximately constant. It is more reasonable to investigate the scalability of a routing protocol by increasing the field size than increasing the node density [15]. We used 40 source-destination pairs in all simulations. The source and destination nodes are randomly selected. Traffic sources are CBR (constant bit-rate). Each source sends data packets of 512 bytes at the rate of four packets per second. The same traffic load is used in all simulations to make the simulation results more comparable. The random way point model [20] was used as the mobility model. We assume that a destination node does not change its cell ID. So during a simulation, a destination node moves randomly in its cell without leaving it. In our future work, we will develop a cell ID management protocol, which traces the cell ID of each node. With this protocol, the above assumption can be released. The maximum moving speed is 10 m/s and the minimum speed is 1 m/s. We varied pause time to change mobility rate. The pause time was 0, 100, 200, 400, 600 and 900 simulated seconds. For each data point, five simulation runs were executed. The final result is the average of these runs. Identical traffic and mobility scenarios are used for both the routing protocols. For AODV, we used the original code in the ns2 package. For ABRP, the TTL of ANCHOR-ANNOUNCEMENT message was 2 hops.

#### C. Results and Discussion

1) Packet Delivery Ratio: Packet delivery ratio is the number of data packets received at the destinations to the number of data packets generated by the CBR sources. Figures 1 and 2 show the variation of packet delivery ratio under different pause time. With the increase of pause time, mobility is lower, routes are less likely to be broken, and so packet delivery...
ratio goes up. From the two figures, it can be seen that ABRP has better packet delivery ratio than AODV. This is especially obvious for large networks. For AODV, the network size has significant influence. For 200 nodes situation (Fig. 1), the packet delivery is about 81-94 percent. But when the network size increases to 1000 nodes (Fig. 2), the packet delivery ratio decreases to 62-75 percent. For ABRP, the packet delivery rate is about 82-98.3 percent for 200 nodes network. It decreases to 74-95 percent when the network size goes up to 1000 nodes. This is explained as follows. In ABRP, each node maintains routes to its neighbor cells for inter-cell routing protocol, and reactively discovers routes to other nodes within the same cell for intra-cell routing protocol. So the length of a route is no longer than the diameter of a cell plus one hop, even if the size of the network increases. Although the length of a path between a source and a destination is longer in a larger network, the average length of routes of a nodes routing table in ABRP does not change much. So it is less likely that a route is broken than in AODV. For small network, this effect is not very obvious, since the length of a path between a source and a destination is relatively short. For large network, it is significant.

2) **Routing Overhead:** We use normalized routing overhead to measure the efficiency of a routing protocol. The normalized routing overhead is the number of routing messages transmitted in the network for each data packet received at the destination. The routing messages includes all the control messages to establish route, such as route requests and route replies. For ABRP, the routing messages also include control packets to set up cells. For a routing packet transmitted over multiple hops, each hop is considered as one transmission. Figures 3 and 4 show the results. When pause time increases, the routing overhead goes down, since mobility is lower and less routes are broken. It can be seen that ABRP outperforms AODV in all the scenarios. Especially for large networks, like 1000 nodes situations, ABRP has 10-40 percent less overhead than AODV. It can be explained as follows. In AODV, a node discovers a route when it has a packet to be sent and does not have the route. For ABRP, the intra-cell routing protocol is also an on-demand routing protocol and route discovery is confined within each cell. For the inter-cell routing protocol, each node maintains routes to all its neighbor cells. Generally, a node has multiple routes to each neighbor cell. A new round of route discovery is needed only when all the routes to a neighbor cell are broken, which can reduce overhead. As discussed in Section II-B, an inter-cell route request naturally establishes an intra-cell routing route to the source node. In general, the intra-cell routing protocol is not necessary to be executed. In large network, the average path length is longer, so for AODV, route rediscovery happens more frequently than in small network, since longer path is more likely to be broken. This increases overhead. But for ABRP, each node only maintains inter-cell routing routes to its neighbor cells and establishes intra-cell routing routes within its cell, so the size of network has little impact on the routing activity of a node. For small network, the average path length is shorter, so the overhead difference between ABRP and AODV is smaller than in large network.

3) **End-to-end Delay:** We calculated this metric as an average over all received data packets. It consists of all possible delays, such as buffering at routing layer, queuing at interface queue, and retransmission at MAC layer. It should be pointed
out that this metric is in favor of lower delivery rate protocols since the value is only calculated over received data packets, which in general go through shorter path and has less delay. From figures 5 and 6, it can be seen that ABRP has less delay than AODV. Even in 1000 nodes situation, in which AODV has low packet delivery rate, ABRP outperforms AODV. For AODV, if a link is broken, either the upstream node of the link does local route repair or the source node does route rediscovery. Either case causes delay. For ABRP, each node in general maintains multiple routes to a neighbor cell, if one route is down, it still has backups. If all the routes to the neighbor cell with the shortest distance to the destination node’s cell are invalid, the node can try the neighbor cell with the second shortest distance to the destination node’s cell. Although these backup routes are not the shortest, the incurred delay is less than the delay caused by route rediscovery.

IV. RELATED WORK

Landmark Ad Hoc Routing Protocol (LANMAR) [21], [22] is a proactive routing protocol. It is designed for ad hoc networks that have coordinated group mobility. Each motion group naturally forms a logical subnet and is identified by a groupID. Each node in the network has a unique identifier, <groupID, nodeID>, where nodeID is the node’s address. In each group, a node is elected as landmark. Within each group, a proactive routing scheme, Fisheye State Routing (FSR) [23], is used to establish routes between nodes in the same subnet. A distance vector routing scheme is used to disseminate the routes to the landmarks of all the subnets. So each node in the network maintains two kinds of routes: one to its nearby destinations and one to the landmarks of all the subnets. Thus a node can directly send packet to other nodes within same subnet. If the destination node is in a different logical subnet, the source node transmits the packet towards the landmark of the destination’s logical subnet. When the packet reaches the destination’s subnet, it can be routed to the destination directly, not necessarily through the landmark node.

Morris et al. [24] proposed a location proxy scheme. They assume that some nodes in the network know their location and other nodes do not know it. The location unknown node picks a nearby location-aware node as its proxy. A modified DSDV routing protocol is used to establish routes between location proxies and location unaware nodes and their proxies. A packet destined to a location-unaware node is first forwarded towards that node’s proxy and then delivered with the modified DSDV routing protocol after the packet arrives at the proxy.

V. CONCLUSION AND FUTURE WORK

In this paper, we presented ABRP, a scalable routing protocol for mobile ad hoc networks. ABRP is a hybrid routing protocol, which combines the advantages of table based routing approach and geographic routing approach, while avoid the burden—GPS support.

We have evaluated and analyzed the performance of ABRP with ns2 simulator. The node density is maintained constant to better evaluate the scalability of the protocol. Simulation results show that when compared with AODV, ABRP has higher packet delivery ratio, lower overhead and low end-to-end delay. Especially for large network, ABRP achieves 25 percent higher packet delivery rate than AODV, with 20 percent lower overhead. This denotes that ABRP has better scalability than AODV.

In this paper we assume that the source node knows the cell ID of the destination node and the destination node does not
change its cell ID during the simulation. In our future work, we will design a cell ID management protocol, which traces the cell ID of each node. With this protocol the above assumption can be removed. The source node obtains the cell ID of the destination node from the cell ID management protocol and put it in to the header of a data packet. When the data packet reaches a bridge node in the destination nodes cell, the bridge node checks if the destination node is still in the cell. If not, the bridge node obtains the new cell ID of the destination node from the cell ID management protocol and forwards the packet to the new cell.

ACKNOWLEDGMENT

This research was partially supported by NSF Grants IIS-0242384 and IIS-0324836.

REFERENCES