Cluster based Routing Protocol for Mobile Ad Hoc Networks

M. Rezaee[1], M. Yaghmaee[2]

Department of Computer Engineering, Ferdowsi University of Mashhad, Iran

[1]mrk3796@gmail.com

[2]myaghmaee@ferdowsi.um.ac.ir

Abstract: In this paper, we proposed a cluster based routing protocol for mobile ad hoc network. It uses clustering's structure to decrease average end-to-end delay and improve the average packet delivery ratio. We simulate our routing protocol for nodes running the IEEE802.11 MAC. Results of our experiments show that the packet delivery ratio increases greatly and packet delay decreases significantly. In proposed method the routing is also done quickly and its error tolerance increases. The reason is that, routing is depended on the address of cluster heads. By failing any node in the route, its cluster head may use another node to forward packets (if available). This causes the error tolerance to enhance.

Keywords: MANET, Routing, Clustering.

(Received September 09, 2008 / Accepted February 23, 2009)

1. Introduction

Mobile ad hoc networks (MANETs) are collection of mobile nodes that intercommunicate on shared wireless channels. The topology of the network changes with time due to mobility of nodes. Nodes may also enter or leave the network. These nodes have routing capabilities which allow them to create multihop paths connecting node which are not within radio range.

The routing protocols can be roughly divided into three categories: proactive (table driven routing protocols), reactive (on-demand routing protocols), and hybrid. The primary goal of such an ad hoc network routing protocol is to provide correct and efficient route establishment between pair of nodes so that messages may be delivered in time.

In proactive, each node maintains a routing table, containing routing information on reaching every other node in the network. In order to have this routing information available and update, routing information must be exchanged. Proactive routing protocol use periodic broadcast to establish routes and maintain them. This can cause substantial overhead (due to the “route message” traffic), affecting bandwidth utilization, and throughput as well as power usage. The advantage is that routes to any destination are always available without the overhead of a route discovery.

In reactive, when a node whishes to send packet to a particular destination, it initiates the route discovery process, in order to find the destination.

Reactive routing protocol provides a cost-effective solution for packet routing. However, when routes are requested, nodes need to send out route query messages into a large part of the network, which could lead to the delay of route response and potentially a large penalty in network resources. This situation causes throughput loss in high mobility scenarios, because the packets get dropped quickly due to unstable route selection.

Hybrid protocols combine the benefit of both approaches. Hybrid protocols are scalable to network size.

In this paper, we propose a new hybrid routing algorithm for MANET called Cluster based Routing Protocol (CRP) algorithm. It uses clustering's structure to decrease routing control overhead and improve the networks scalability.

Results of our simulations show that the packet delivery ratio increases greatly and packet delay decreases significantly, when compared with other routing algorithms such as ad hoc on-demand Distance Vector (AODV).

2. Related work

Many routing algorithms have been proposed in the past. Destination Sequenced Distance Vector (DSDV) [1] is a proactive routing protocol. It is based on classical Bellman ford routing algorithm for finding shortest paths between ad hoc nodes with some improvement. DSDV is a hop-by-hop distance vector routing protocol requiring each node to periodically broadcast routing updates. The key advantage of DSDV over traditional distance vector protocols is that it guarantees loop-freedom.

Optimized Link State Routing (OLSR) [2] protocol is a proactive routing protocol. The key concept used in the protocol is that of multipoint relays (MPRs). MPRs refer to selected routers that can forward broadcast messages during the flooding process. The idea of multipoint relays is to minimize the flooding of broadcast packets in the network by reducing duplicate retransmissions in the same region. Although OLSR provides a path from source to destination, it is not necessarily the shortest path, because every route involves forwarding through a MPR node. A further disadvantage is that OLSR also has routing delays and bandwidth overhead at the MPR nodes as they act as localized forwarding routers.

The Dynamic Source Routing (DSR) [3] protocol is a reactive routing protocol for MANETs. The key feature of DSR is the use of source routing, which means the sender knows the complete hop-by-hop route to the destination. A complete list of intermediate nodes to the destination kept in the header of each data packet. Scalability and poor performance in high mobility and heavy traffic loads are disadvantage of DSR, because DSR relies on blind broadcasts to discover routes.

AODV [4] is essentially a combination of both DSR and DSDV. It borrows the basic on-demand mechanism of Route Discovery and Route Maintenance from DSR, plus the use of hop-by-hop routing, sequence numbers, and periodic beacons from DSDV. AODV is loop-free due to the destination sequence numbers associated with routes. It creates routes only on-demand, which greatly reduces the periodic control message overhead associated with proactive routing protocols. Similar to DSR, poor scalability is a disadvantage of AODV.

Bypass-AODV [5], an extension of AODV, uses a specific strategy of cross-layer MAC-interaction to identify mobility-related link breaks, and then setup a bypass between the broken link end nodes via an alternative node. By restricting the bypass to a very small topological radius, routing overheads are minimized considerably.

The Temporally-Ordered Routing Algorithm (TORA) [6] is the most well known LRR (Link Reversal Routing) algorithm which provides a very adaptive type of routing. It uses a mix of reactive and proactive routing. Sources initiate route requests in a reactive mode. At the same time, selected destinations may start proactive operations to build traditional routing tables.

Zone Routing Protocol (ZRP) [7] is a hybrid routing algorithm. In ZRP, the network is divided into zones. A proactive table driven strategy is used for establishment and maintenance of routes between nodes of the same zone, and a reactive on-demand strategy is used for communication between nodes of different zones. When a destination is out of the zone, on-demand routing search is initiated. In this situation, control overhead is reduced, compared to both the route request flooding mechanism employed in on-demand protocols and periodic flooding of routing information packet in table driven protocol.

Virtual Backbone Routing (VBR) [8] is a scalable hybrid routing framework for ad hoc networks, which combines local proactive and global reactive routing components over a variable-sized zone hierarchy. It is based on the ZRP's concept of combining proactive local zone routing with reactive global route queries. However, unlike ZRP, VBR utilizes the notion of a virtual backbone (VB) to efficiently direct the route querying control traffic. The reactive component of VBR restricts the route queries to within the virtual backbone only, thus improving the overall routing efficiency.

3. Clustering

We use clustering's structure for routing protocol.

Clustering is a process that divides the network into interconnected substructures, called clusters. Each cluster has a cluster head (CH) as coordinator within the substructure. Each CH acts as a temporary base station within its zone or cluster and communicates with other CHs.

In our protocol, there are four possible states for the node: NORMAL, ISOLATED, CLUSTERHEAD and GATEWAY. Initially all nodes are in the state of ISOLATED. Each node maintains the NEIGHBOR table wherein the information about the other neighbor nodes is stored

CHs have another table (CHNEIGHBOR) wherein the information about the other neighbor CHs is stored.

The primary step in clustering is the CH election.

3.1. Election algorithm

In this algorithm we allocate weight to every node. The weights are in three groups, that each group specifies the credit measure of node to becoming head. Nodes will send a message to that head and each node that is in the higher group and delivers the message to neighbor nodes sooner would be chosen as the CH. This work makes the head node choosing in a less time.

Each node periodically broadcasts LIVE message to declare itself and to have knowledge of its neighbor nodes, which can be used to calculate its W parameter. W specifies weight group. W is calculated from weight. Each node periodically calculates the weight.

$$Weight = aN + bR + cT + dP \qquad (1)$$

The N is the number of neighbors of node. Node does not add to N until it is certain that the power level of the last received messages from the neighbor is greater than the power level of the first received message. In this way, Node is sure that the neighbor is moving closer to it. The R is remaining battery lifetime of node (percentage of remaining over full battery power) of node. It allows extending the lifetime of nodes by relinquishing the role as a CH in case of insufficient battery power. The T is the cumulative time during which the node had been in the last cluster. T implies node stability thus, increases the stability of cluster. The P is transmission power and used in order to elect the node which can cover the largest range, thus, minimizes the number of generated clusters.

The coefficients a, b, c and d are weighting factors for the corresponding system parameters and a + b + c + d=1. The flexibility of changing the weight factors helps us apply our algorithm to various networks.

Each node builds its NEIGHBOR table based on the received information.

Nodes can start to elect CH after the predetermined period of time (T_e). If an ISOLATED node before Te receives LIVE message from any CH, sets ID_{ch} with CH address and sets its state to NORMAL. If ISOLATED node dose not receive message from any CH after Te time, it searches the NEIGHBOR table for nodes with higher W parameter than its W parameter. If node does not find node in neighbor table with higher W parameter than its W parameter, it elects itself as CH and sets its state to CLUSTERHEAD and set ID_{ch} with its own ID. Otherwise it continues to send LIVE message until $2T_e$ time. If a node after 2Te time does not receive LIVE message from any CH, it declares itself as CH and sets it's STATE to CLUSTERHEAD and set ID_{ch} with its own ID.

When the NORMAL node receive LIVE message from another CH and do not find it in its NEIGHBOR table, it sends CTGATEWAY message to its CH and insert CH information in its NEIGHBOR table. The CTGATEWAY message includes information of new CH (X). CH by receiving CTGATEWAY message from normal node checks the CHNEIGHBOR table and if does not find X, it inserts X information in the CHNEIGHBOR table and sends ARGATEWAY message to normal node. Otherwise sends NRGATEWAY message to normal node. When NORMAL node receive ARGATEWAY message from CH, it goes to GATEWAY state and sets its ID_{ch} with highest W CH. NORMAL node for changing to GATEWAY state requires to receive accept from its CH, to prevent from creating unused GATEWAY node and reduce incremental overhead.

If NORMAL node receives LIVE message from another NORMAL or GATEWAY node in a different cluster, it repeats up the task (Figure 1).

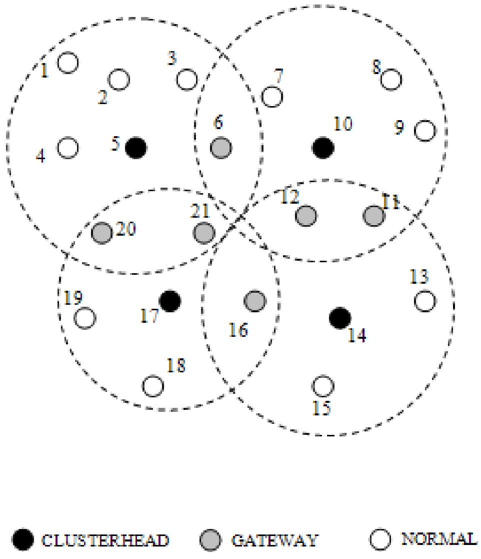

Figure 1: cluster formation

3.2. Cluster maintenance

Because of the nodes mobility, the network topology will change over time. A node may join or leave an existing cluster at a time. Two CHs may come within one hop, which may trigger a cluster head change event. Unfortunately, the moment that two CHs hear LIVE message from each other, it may be frequently due to rapid node mobility in mobile ad hoc networks.

In our algorithm, for the two meeting CHs A and B, if A is the first one to receive LIVE massage, it checks NEIGHBOR table. If all member nodes were GATEWAYs it changes the state to a NORMAL and sets ID_{ch} with address of B. Otherwise it checks its W parameters. If A finds out that the W parameter is lower, it simply changes the state to GATEWAY and set ID_{ch} with address of B. The reason to convert to GATEWAY node is because when an existing CH gives up the cluster head role, where the member does not range B can use A for service (e.g. routing).

If A finds that it has higher W parameters, it sends a unicast COVERLAP message to B. Then B terminates its clusterhead role and changes the state to GATEWAY and sets ID_{ch} with the address of A.

4. Routing

When a source node S wants to deliver data to an unknown (no match in routing table) node D, S first check its neighbor table, if there is a match, it simple adds this route into routing table and directly send data to D. otherwise s initiate a path discovery process to locate the destination.

Each node maintains the routing table where in the information about the routes is stored. The format of this table is defines as: RTABLE (ID_{dest}, ID_{source}, ID_{next}, DSN, HC, and LT). Here, ID_{dest} is destination address, ID_{source} is source address, ID_{next} is next hop node address, HC is hop count and LT is route life time.

The destination sequence number (DSN) field is the last known destination sequence number for each destination and is copied from the destination sequence number field in the routing messages. The DSN is used to distinguish stable routes from new ones to avoid the formation of loops. The DSN is incremented every time that the source node initiates a route request message.

4.1. Route discovery

When the source node wants to send a message to the destination node and does not already have a valid route to that destination, it initiates a path discovery process to locate the destination.

When a source node S seeks to set up a connection to a destination D, S send route request message (RDemand) to its cluster head. RDemand message includes the following fields:

ID_{dest}, ID_{source}, DSN, HC, and LT.

It may be the case node D may be available within the cluster or across the cluster. If node D falls within the cluster or cluster head has a valid route to the destination node, then the cluster head sends RResponse message to the node S. Otherwise, S forwards the RDemand message to the cluster heads available in its CHNEIGHBOR table and updates the its routing table.

When an intermediate CH node receives the RDemand from its neighbors, it first increases the hop count value in the RDemand, to account for new hop through the intermediate node if the packet should not be discarded.

The originator sequence number contained in the RDemand must be compared to the corresponding destination sequence number in the route table. If the originator sequence number of the RDemand is not greater than existing value, it the intermediate node discards it. If the originator sequence contained in the RDemand is greater than the existing value in its route table, the relay CH creates new entry with the sequence number of the RDemand.

Once the RDemand has arrived the destination CH or an intermediate CH with an active route to destination, node forwards the message of finding the route in the reverse route (RRespone message). To prevent any rooting loop, any routr discovering message has a number which with the association of the beginning id, produces a unique number. Source node by receiving the RRespone from the destination node; update its routing table and starts forwarding packets.

4.2. Route maintenance

The routing table is created due to the address of cluster heads. This means that, in routing tables, the address of the next CHs are saved for any destination. In the previous methods if a node failed within a route or become far from its neighbor nodes, it causes the route to fail and leads to the recreating of path. But in the proposed method, since the route is expressed due to the CHs, in case of the fail of a node in a route, the CH of that node can use another node to forward a packet to the next existed in the route (Figure 2). In this method, only when a cluster fails or corrupts the needs for the recreating of the path arises, which regarding the attempt for creating more stable clusters, happens less often.

When a CH node detects a link break for the next hop CH of active route, it sends a route error packet (RERR) back to all precursors. The format of this message is RERR (ID, SEQNUM). Here, ID is the destination address and SEQNUM is destination sequence number. When a CH node receives a RERR from a neighbor CH for one or more active route, it forward the packet to precursors stored in its route table. When a source node receives a RERR, it initiates a new route discovery if the route is still needed.

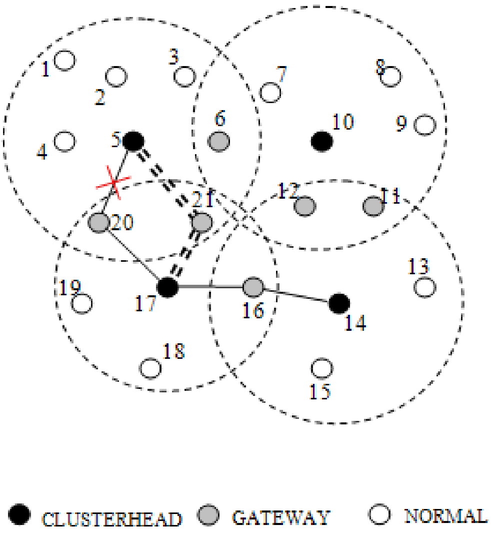

Figure 2: Local route repair

5. Simulation

In this section, we evaluate the proposed routing protocol and compare it with AODV routing protocol via simulation. For this purpose, we implemented the proposed algorithm on the NS-2 [9] simulator. The performance of the CRP and AODV protocol is evaluated in terms of packet delivery ratio and average end-to-end delay. The packet delivery ratio is defined as the percentage of packets that successfully reach the receiver nodes each second. The average end-to-end delay is defined as the average time between a packet being sent and being received.

The scenarios were generated with input parameters as listed in table 1. The number of mobile nodes is set to 50 to 300 nodes. These nodes are spread randomly in a 500m X 500m area network.

The random waypoint model is used to model mobility. Each node starts its journey from a random location to a random destination point with a specific speed. Once the destination is reached, another random destination point is targeted after a pause time.

Table 1 Simulation parameters

Parameter	Value
Number of nodes	50-300
Size of network	500m x 500m
Speed of the nodes	5-20Km/h
Transmission rang	30-300 m

Figure 3 compares the packet delivery ratio (PDR) for AODV and CRP. As the number of nodes increases the packet delivery ratio decreases. We can see that the packet delivery ratio of CRP is clearly higher than the AODV protocol and our algorithm can scale up to larger network.

The comparison of the end-to-end delay is show in Figure 4. We can see that as the total number of nodes increases, the average end-to-end delay increases, because more connections and congestions appear in higher density network. It can also be concluded from this study (Figure 4) that the average end-to-end delay for proposed approach is better than the AODV protocol.

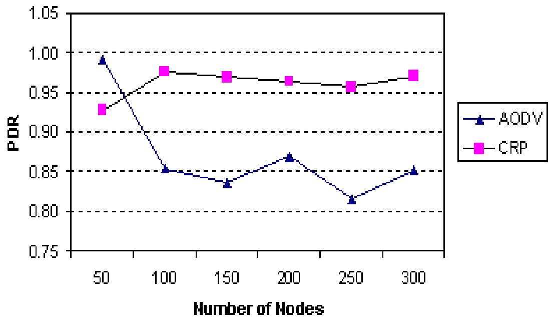

Figure 3: PDR vs. number of nodes

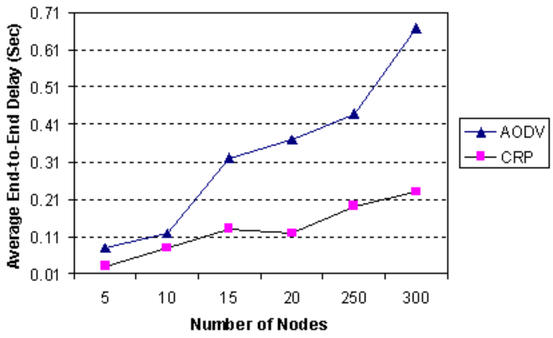

Figure 4: Average end-to-end delay vs. number of nodes

Figure 5 shows the comparison of packet delivery ratio for AODV and CRP in different mobility. As the velocity of nodes increases, the probability of link failure increases and hence the number of packet drops also increases. However, the delivery ratio of CRP drops down more slowly, which means that our method has better performance.

Figure 6 shows the comparison of the average end-to-end delay for AODV and CRP in various mobile velocities. For both CRP and AODV routing, it can be seen that there is increases in average end-to-end delay with the increases of node speed. CRP routing provides smaller packet delay than AODV. This is because CRP routing protocol has local path repair and need smaller route discovery time.

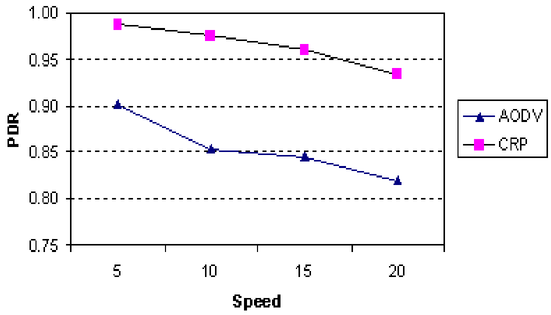

Figure 5: PDR vs. node speed (50 nodes)

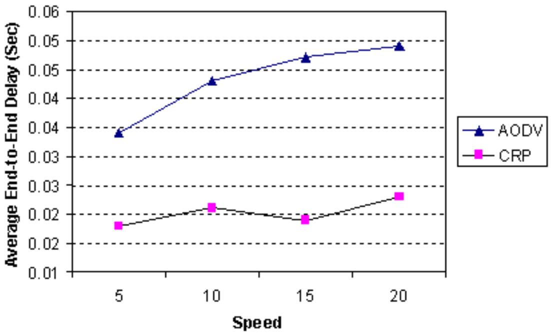

Figure 6: Average end-to-end delay vs. node speed (50 nodes)

Figure 7 compares the average end-to-end delay in various pause times. With the decreases of pause time, the average end-to-end delay for both CRP and AODV routing increases. The reason is that the network topology changes more frequently at smaller pause time.

Figure 8 shows the comparison of the packet delivery rate and pause time. As mobility rate increases, the packet delivery rate decreases due to link breaks. The CRP performs well when compared to that of AODV.

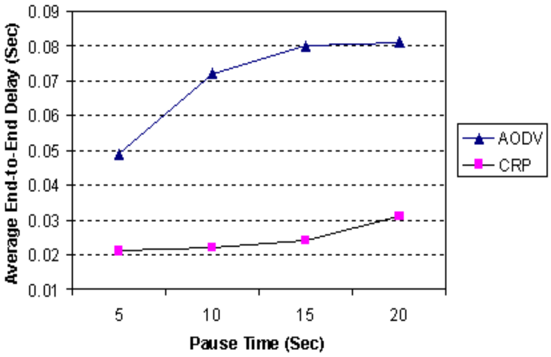

Figure 7: Average end-to-end delay vs. pause time (50 nodes)

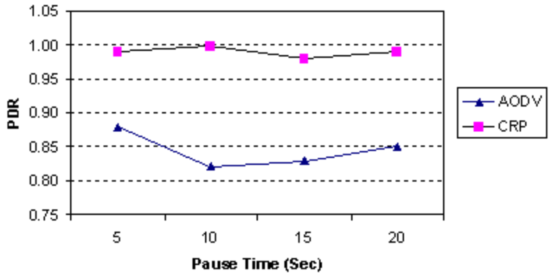

Figure 8: PDR vs. pause time (50 nodes)

6. Conclusion

Our proposed algorithm is a cluster based routing protocol for ad hoc network. In our method, due to the weight group, the cluster creation speed increases, and causes the network services to be more accessible. Recreating of clusters is rarely executed, and when two clusters locate in the same range, one of them becomes the gateway of other node. This causes to prevent the creation of most constructions. In the proposed protocol the routing is also done quickly. The reason is that, routing is depended on the address of cluster heads. By failing any node in the route, its CH may use another node to forward packets (if available). This causes the error tolerance to be enhanced. The performance of proposed protocol has been evaluated through extensive simulation with network topologies of various sizes. Simulations demonstrate significant improvements in packet delivery ratio over traditional routing protocol and better performance than other routing algorithms in literature as well.

References

[1] Perkins, C.E. and Bhagwat, P. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, Comp. Commun. Rev, pp. 234-44, 1994.

[2] Clausen, T. and Jacquet, P. Optimized link state routing (OLSR) protocol, RFC 3626, IETF, 2003.

[3] Johnson, D.B. and Maltz, D.A. Dynamic source routing in ad-hoc wireless networks, Mobile Computing, T. Imielinski and H. Korth, Eds., pp. 153-81, 1996.

[4] Perkins, C.E. and Royer, E.M. Ad-hoc on-demand distance vector routing, Proc. IEEE Workshop on Mobile Computing Systems and Applications, pp. 90-100, 1999.

[5] Alshanyour, M. and Baroudi, U. Bypass AODV: Improving Performance of Ad Hoc On-Demand Distance Vector (AODV) Routing Protocol in Wireless Ad Hoc Networks, Ambi-sys, 2008.

[6] Park, V.D. and Corson, M.S. A highly adaptive distributed routing algorithm for mobile wireless networks, Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies, pp.1405-1413, 1997.

[7] Haas, Z.J. and Pearlman, M.R. Zrp: a hybrid framework for routing in ad hoc networks, pp. 221-253, 2001.

[8] Liang, B. and Haas, Z.J. Hybrid Routing in Ad Hoc Networks with a Dynamic Virtual Backbone, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, VOL. 5, NO. 6, 2006.

[9] Network Simulator 2, http://www.isi.edu/nsnam