Abstract

This paper describes improvement in standard routing protocol AODV for mobile ad-hoc networks. Our mechanism setups multiple paths based on less hop count. It allows to store multiple paths based on threshold. At time of link failure, it will switch to next available path. To set up multiple path, we have used the information that we get in the RREQ packet and also send RREP packet to more than one paths. It reduces overhead of local route discovery at the time of link failure. And because of this End to End Delay and Drop Ratio decreases. The main features of our mechanism is that it is simple, efficient. We evaluate through simulations the performance of the AODV routing protocol including our scheme and we compare it with the basic AODV and multipath extension of AODV-AOMDV routing protocols. Results show that our new concept outperforms the basic AODV. Indeed, our scheme reduces routing load 10% for AODV and 3% in AOMDV and decreases the End to End delay and Drop Ratio.

1. Introduction

A Mobile Ad hoc NETwork (MANET) is a kind of wireless ad-hoc network, and is a self-configuring network of mobile routers (and associated hosts) connected by wireless links – the union of which forms an arbitrary topology. The routers are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion, or may be connected to the larger Internet. [1]

Moreover, mobile platform moves autonomously and communicates via dynamically changing network. Thus, frequent change of the network topology is a main challenge for many important topics, such as routing protocol, robustness and performance degradation [2]. For MANET, various routing protocols are available each has its own characteristics and some of them have derived characteristics. Depending upon the nature of application, appropriate routing protocol is implemented. Proactive and reactive protocols are the two classes of MANET routing protocols and each constitute a set of protocols. In this paper we consider the Ad-hoc On-Demand Distance Vector Routing Protocol (AODV) that uses a demand-driven route establishment procedure. The Ad-hoc on demand distance vector (AODV) routing algorithm is a routing protocol designed for Ad-hoc mobile devices. AODV is Combination of DSR and DSDV. It has on-demand mechanism of route discovery and route-maintenance from DSR. Plus the hop-by-hop routing, sequence numbers and periodic beacons from DSDV.

Lots of modifications have been done in original AODV protocol. XIAO Bai-Long, GUO Wei, LIU Jun and ZHU Si-Lu have presented an Improvement in Local Route in Mobile Ad Hoc Networks, in which repair is confined to the vicinity of the broken link to decrease the reaction time of recovery and the overhead of route maintenance.[3]

Kokula Krishna Hari has proposed and implemented a different mechanism and a scheme to make the important time critical data like real time or voice data to reach the destination without any loss. The scheme used is temporary path between the nodes during link failure. The important node then forwards the buffered packets to the destination without any loss which is on-demand based on type of information a
node forwards. Special buffer is allocated for the nodes marked important during special propagation.[4]

Mehdi EffatParvar, MohammadReza EffatParvar, Amir Darehshoorzadeh, Mehdi Zarei, Nasser Yazdani have presented LBAODV which is a new multipath routing protocol that uses all discovered path simultaneously for transmitting data, by using this approach data packets are balanced over discovered paths and energy consumption is distributed across many nodes through network. [5]

Here we have considered an multipath version of this algorithm namely AOMDV proposed by Mahesh K. Marina and Samir R. Das. AOMDV computes multiple loop-free link disjoint paths during route discovery. They introduce the notion of an advertised hopcount to establish loop free

The remainder of this paper is organized as follows. In Section 2, we give the most important characteristics of AODV routing protocol. In Section 3, we describe multipath extension of AODV-AOMDV. Section 4 explains our algorithm in detail. Simulation methodology and performance evaluation of our proposal are detailed in Section 5. Section 6 concludes the paper by summarizing results

2. Basic AODV routing protocol

Basically working of Ad hoc On demand Distance Vector (AODV)[6] routing protocol can be divided into two phase:

2.1 Route Discovery Phase.

When a node has data to send to a node for which it does not have route, Route discovery phase is started. It will broadcast RREQ packet. Each intermediate node will cache the route back to the originator (Reverse path setup), so that RREP can be unicast back to the source.

After receiving RREQ packet, destination node unicast RREP packet to the originator. Every intermediate node receiving RREP, will establish a route to the destination (Forward path setup). If an intermediate node has a fresh enough route for the desired destination, it will respond to the RREQ.

2.2 Route Maintenance

When a link breakage in an active route occurs, the upstream node of that break may choose to repair the link locally. To repair the link break, it initiates route discovery for the destination. If, at the end of the discovery period, the repairing node has not received a RREP (or other control message creating or updating the route) for that destination, it transmits a RERR message for that destination to the source.


Ad-hoc On-demand Multipath Distance Vector (AOMDV) [7] Routing protocol is a multipath extension to the AODV protocol which maintains multiple loop-free and link disjoint paths. The routing entries for each destination contain a list of the next-hops and corresponding hop counts. All the next hops have the same sequence number. This helps in keeping track of a route. For each destination, a node maintains the advertised hop count, which is the maximum hop count for all the paths, which is used for sending route advertisements of the destination. Each duplicate route advertisement received by a node defines an alternate path to the destination.

Loop freedom is assured for a node by accepting alternate paths to destination if it has a less hop count than the advertised hop count for that destination. Because the maximum hop count is used, the advertised hop count therefore does not change for the same sequence number [8]. When a route advertisement is received for a destination with a greater sequence number, the next-hop list and the advertised hop count are reinitialized. AOMDV can be used to find node-disjoint or link-disjoint routes.

To find node-disjoint routes, each RREQs arriving via a different neighbor of the source defines a node-disjoint path. This is because nodes cannot broadcast duplicate RREQs, so any two RREQs arriving at an intermediate node via a different neighbor of the source could not have traversed the same node. [9]

In an attempt to get multiple link-disjoint routes, the destination only replies to duplicate RREQs arriving via unique neighbors. After the first hop, the RREPs follow the reverse paths, which are node disjoint and thus link-disjoint.

The trajectories of each RREP may intersect at an intermediate node, but each takes a different reverse path to the source to ensure link disjointness [8]. The advantage of using AOMDV is that it allows intermediate nodes to reply to RREQs, while still selecting disjoint paths. But, AOMDV has more message overheads during route discovery due to increased flooding and since it is a multipath routing protocol, the destination replies to the multiple RREQs those results in longer overhead.
4. Proposed System

In the proposed algorithm, multipath is discovered and maintained in advance at the time of route discovery, but instead of considering each and every RREQ at each node it will consider only specified number of request (based on REQ_CNT_THRESHOLD). At destination or intermediate node, RREP is sent to every received RREQ from unique node. Thus more than one path is maintained all with same but least hop count will be stored in routing table and one of them will be used for data transfer. Other non used paths will be used at time of link breakage.

Proposed algorithm has three phases, Route Discovery, Data Sending and Route Maintenance

4.1 Route Discovery

Route discovery is initiated by the source node when it has some data to send and does not have the route table entry for the destination. It broadcasts RREQ packet to its neighbors.

When Intermediate node gets RREQ, it will check for the route table entry, for the destination mentioned in the RREQ packet. If it finds route table entry for the destination it will generate RREP packet and send it to the source. If it doesn’t have the route table entry for that destination it will rebroadcast the RREQ, after updating the route entry for the source. It will consider REQ_CNT_THRESHOLD number of RREQ packet with same (least) hop count for the one request at each node and for that it maintains one counter for each RREQ.

When RREQ packets come at the destination, it will generate RREP packet for each RREQ packet, and unicast it to the source. At each intermediate node, route to the destination will be established by recording the next hop to the destination.

4.2 Data Sending

Data will be sent as soon as the first RREP packet comes to the source data packets will be sent and it will traverse hop by hop.

4.3 Route Maintenance

If a link break is detected, it will check for the unreachable destination and if any, it will broadcast a Route Error (RERR) packet. All the node getting RRER packet, will re broadcast it if and only if there is at least one unreachable destination.

As we have alternate paths, when data packets arrives, it will use the next path which is available. i.e. It switch to the next path on route failure and will send the RERR only when it does not have any alternate path for the destination.

5. Performance Evaluation

Implementation of wireless ad-hoc networks in the real world is quite hard. Hence, the preferred alternative is to use some simulation software which can mimic real-life scenarios. Though it is difficult to reproduce all the real life factors, most of the characteristics can be programmed into the scenario.

5.1 Methodology

To compare on-demand ad-hoc routing protocol, it is best to use identical simulation environments for their performance evaluation.

5.1.1 Simulation Environment. We make use of ns-2.34 which has support for simulating a multi-hop wireless ad-hoc environment completed with physical, data link, and medium access control (MAC) layer models on ns-2. The protocols maintain a send buffer. It contains all data packets waiting for a route, such as packets for which route discovery has started, but no reply has arrived yet. All packets sent by the routing layer are queued at the interface queue till the MAC layer transmits them. Routing packets get higher priority than data packets.

Our evaluations are based on the simulation of 50 wireless nodes forming an ad hoc network, moving about over an area of (1500 x 300m) flat space for 900s of simulated time. To enable fair and direct comparisons between the routing protocols, identical loads and environmental conditions had to be maintained. Each simulator run accepts an input scenario file describing the motion of mobile nodes and also the sequence of packets originated by the mobile node, along with time of change in motion or packet origination pattern.

5.1.2 Movement Model. In the simulation, node movement is according to random waypoint model. The scenario files used for each simulation are characterized by different pause times. Each mobile node begins the simulation by remaining stationary for the pause time duration. On expiry of pause time, the node chooses a random destination in the 1500m x 300m simulation space and moves there at a uniform speed. Upon reaching the destination, the mobile node pauses again, selects another destination and proceeds there. This
behavior is repeated for the entire duration of the simulation. We ran the simulation with movement patterns generated for 10 different pause times: 0, 100, 200, 300, 400, 500, 600, 700, 800 and 1000s. A pause time of 0 seconds correspond to continuous motion and a pause time of 900s (the length of the simulation) corresponds to no motion.

5.1.3 Communication Model. Here CBR traffic sources are used. A traffic generator named cbrgen is developed to simulate constant bit rate sources in network simulator version 2 (ns2.34). The number of source destination pairs is varied to change the offered load in the network. 30, 35 and 40 pair of UDP streams are used and each CBR packet size is 512 bytes.

5.2 Performance Metrics

We compared the performance of AODV, AOMDV and MAODV by the following three metrics.

5.2.1 Drop ratio. The ratio of the data packets dropped to those generated by the CBR sources.

\[
\text{Drop Ratio} = \frac{\sum \text{CBR dropped}}{\sum \text{CBR sent}}
\]

Where n is number of dropped packets and m is number of sent packets.

5.2.2. Average End to end data delay. This is average end to delay of all successfully transmitted data packets from source to destination. Formula for average end to end delay is:

\[
\text{Avg. End to End Delay} = \frac{\sum n(\text{CBR sent time} - \text{CBR received time})}{\sum n \text{CBR received}}
\]

Where n is number of received packets.

5.2.3 Normalized Routing Load(NRL). NRL is number of routing packets transmitted per data packet delivered at the destination. Formula for NRL is:

\[
NRL = \frac{\sum k \text{Routing packets}}{\sum n \text{CBR received}}
\]

Where n is number of received packets, and k is number of routing packets.

6. Results And Analysis.

6.1 Drop Ratio

Figure 1, 2 and 3 shows the relationship between Drop Ratio and Pause time. Pause time indicates mobility.

In case of higher mobility, AOMDV performs better than AODV. It is because at the higher mobility, link breaks occur more frequently, and so multiple stored path can be used.

While in case of lower mobility, AODV performs better than AOMDV. Theoretically it should not be the case, i.e. in stable network also AOMDV should perform better or equal than AODV.

However it can be seen from the graph that Improved AODV performs better than AODV and AOMDV because it has more path than AOMDV.

In general with increase in pause time, Drop ratio should be decreased. But we can see three unusual spikes in all three graphs for all three protocols, i.e. 100, 400, 600 and 800. It is because of random unpredictable behavior of the node.

Further it can be seen from the graph than behavior of all three protocols are same.
6.2 Average End to End Delay.

Fig 4 and 5 and 6 show end-to-end average delay for 30, 35 and 40 connections respectively. As seen in all the plots the end-to-end average delay is continuously decreasing with increase in pause time for all basic AODV AOMDV and Improved AODV till some point.

The reason behind that is at lower pause time, the failure of the route recovery attempts are more therefore the number of packets transferred between the source and destinations are also more. It causes more time taken to recover the route at lower pause time. We find that end-to-end average delay is consistently lower in Improved AODV than basic AODV and AOMDV protocol.

Again for higher pause time, the behavior is opposite to the theory. The reason is because of random position of the nodes. Typically with pause time 800, we don’t get expected result, because of randomness of the nodes.
6.3 Normalized Routing Load

Fig. 7, 8 and 9 demonstrates routing load in presence of different number of connections with varying pause time.

The number of connections will affect the requirement of route discovery between different pairs of source and destination, in addition to traffic on the MANET.

The pause time indicates the mobility of the nodes. This in turn shows increase in routing load increases with increase in number of connections.

It can also be observed in the fig, routing load is continuously decreasing with increase in pause time in all three conditions for all three cases.

This is understandable, as increase in pause time indicates reduced mobility, which in turn reduces the requirements of route discovery. At lower pause time, (pause time 0) link layer reports high route failure to its upper layer.

7 Conclusion.

The proposed algorithm will generate slightly higher overhead than that of AODV for first time at the time of route discovery. But once route discovery is over, it will be beneficial for route maintenance. And this overhead overcomes the route overhead generated at the time of link failure. i.e. Normalized routing overhead for Improved AODV will be lower than that of AODV and AOMDV.

Normalized Routing Load is 10% decreased in AODV and 3% in AOMDV.

End to end delay of Improved AODV is 1/5th of AODV and AOMDV as there would not be any delay because of link failure and also because data will be sent as soon as first RREP would be received.
Drop ratio for AODV is 1.48 times more than Improved AODV and that for AOMDV is 1.42 times more than Improved AODV.

8. Future Work

As a future work, We would like to modify it further to improve Packet Delivery Ratio and Throughput. We would also like to compare the protocol with TCP traffic and if needed, we will do further modification to improve its performance.

REFERENCES

[1]. Darshan Singh, Kali Rai, “Performance Based Comparative Analysis of AODV, DSR and DSDV Protocols w.r.t UDP and TCP” thesis submitted in partial fulfillment of the requirements for the award of degree of Masters of Engineering, Thapar University Patiala July 2009
[10] www.isi.edu/nsnam/ns/