Condition aware robust routing algorithm with cross layer technique for ad hoc situations

M. Rajabzadeh\(^a\), F. Adibniya\(^b\), M. ghasemzadeh\(^c\)

\(^a\) Academic Staff of Islamic Azad University, Chalous, Iran, rajabzadeh.ad@gmail.com
\(^b\) Assistant Professor in Yazd University, Yazd, Iran, f.adib@yazduni.ac.ir
\(^c\) Assistant Professor in Yazd University, Yazd, Iran, m.ghasemzadeh@yazduni.ac.ir

Abstract

In a MANET, mobile nodes must cooperate to dynamically establish routes using multi-hop wireless links. Some of the main challenges in this area related to routing protocols. These networks do not require any existing infrastructure or central administration. To make MANETs adaptive to different mobility and traffic patterns, this paper proposes a novel routing scheme which is utilized mobile agents and attempts to develop DSR protocol in MANETs with simple node-level management behavior resulting in overall system optimization. We develop a probabilistic multi-path routing algorithm and incorporates factors like signal strength into the route metrics so as to predict link breaks before they actually occur. In addition to signal strength and shortest-path metrics, our algorithm updates the goodness of choosing a particular path based on congestion measurement and energy level in each node. By All of the above improvements in DSR, the simulation results testify that our new Robust algorithm can be suitable in ad-hoc situations. We refer to the protocol as the Multi Agent based Adaptive DSR (MA-DSR). Finally, this paper proposes a new cross layer approach and some modifications in the MAC layer for RERR Analysis. By this way, we can detect reason of RERR reception and apply proper mechanisms for performance improvement. We introduce yellow, red and black lists for decrease escaped, low energy, congested and insecure nodes chances in routing strategy. We expect that new condition aware MA-DSR, shows better performance.

Keywords: Condition Aware; Mobile Ad Hoc; MA-DSR; Routing.

1. Introduction

In recent times, new mobile and wireless networks types emerged, offering novel and interesting capabilities. Among them, MANETs represent an immediate example. In a MANET, nodes move arbitrarily [1]. Mobility presents a fundamental challenge to routing protocols. By routing packets cooperatively among the nodes, these nodes can communicate with each other without any central administration [2]. Many routing protocols have been evolved in recent two decades, but most of these protocols only attempt to solve one challenge – Effective Routing – and provide algorithms for minimal end-to-end delay, maximum throughput, etc [3]. Most of the proposed routing protocols tend to use methods that are restricted to network or/and MAC layers in order to deal with the instability of the mobile ad hoc systems. However, their performance in providing better throughput is not satisfactory. There are three reasons for that:

- the sensitivity of algorithms to the degree of nodal mobility.
- the variability of nodes environment.
- the traffic load in mobile ad hoc networks.
nor of them haven’t all of the Optimized routing protocol properties for MANETs. This Properties divided into two major classes: qualitative and quantitative [4]. Qualitative properties consist of Distributed operation, Loop Freedom, Demand Based Operation, Security, Sleep period operation, and Unidirectional link support. But End-to-End data throughput, Delays, Route Acquisition time, percentage of Out of order delivery and Efficiency are some of the Quantitative properties [5]. Hence, suitable improvement in MANET routing protocols acquired when they combined with condition aware traits. After studying some related works in next section, we use theses sextet factors and represent their role in our protocol, step by step. Then we integrate results from each step to reach a complete algorithm for routing problem.

2. Routing Algorithm

To progress the DSR Algorithm some changes should be done in its function. In this chapter we describe this subject completely and analyze the applied changes carefully in route detection and maintain processes.

In proposed MA-DSR Algorithm we rely on 4 basic principles:

- To keep the positive abilities of standard DSR Algorithm.
- Determination of congestion in level of each node.
- Load balancing in network.
- Prediction of links statuses by considering to the received signal power in intermediate and destination nodes to prevent the effects due to link breakage.
- Energy level of each node.

2.1. Congestion in each node

By measuring particular parameters in each node, we obtain the value of congestion in that node. These parameters are including of:

1) Fullness Value of queue (buffer) in each node.
2) Value of load on the channel.
3) Rate of packet dropping.

\[ NC = F(QN, CL, DR) \] \hspace{1cm} (1)

Determinations of F function per above three parameters is the important issue that will also affect on algorithm performance. Here F applies multiplication on the parameters.

\[ NC = (QN \times CL \times DR) \quad 0 \leq QN, CL, DR \leq 1 \] \hspace{1cm} (2)

According to the above calculations value of NC is placed between zero and one that its higher value is indicated the higher congestion. For example, if NC=1, node is in maximum congestion status. Such nodes have lesser chance for cooperation in routing.

2.2. Prediction of links statuses

By comparing the power of received signal in a receiver with transmitted power of sender, the status of communication links would be predictable. Thus, whatever ratio of received power from signal is less than transmitted power, it is indicates that signal attenuation is increasing and by considering to this equation: 

\[ L = 10 \times \log \left( \frac{4 \pi d}{\lambda} \right)^2 \] , the nodes are get away from each other.

\[ SP = \frac{P_1}{P_0} \] \hspace{1cm} (3)

That the initial value is considered 0.5.
2.3. Energy Status in each Node

The limited battery power characteristic limits the utilization time of wireless devices and the networks. Thus, the algorithms and mechanisms that implement the networking functions should be optimized for lean power consumption, so as to save capacity for the applications while still providing good communication performance [6]. In this work, we consider the residual battery power capacity of nodes to reach better performance. From the energy efficiency point of view, we use NE as the parameter to illustrate energy level in each node.

2.4. Final status of each node

By considering to parts of 3.1, 3.2 and 3.3, we calculate final status of each node by the following experimental formula and call it “Node Status Value”.

\[
\text{NSV} = \left(\text{SP} \times \frac{W_1 \cdot (\text{NE}) + W_2 \cdot (\text{NC}) \cdot ^{-1}}{W_1 + W_2}\right)
\]

W1 and W2 are two weight factors to calculate the cost. We can change these two weight factors to change the importance of the two cost metrics during route discovery. For example, we can set (W1, W2) = (0, 1), which means we only consider congestion. Hence, we can scale these factors to change our route selection scheme. In this scheme, W1=W2=1 and W_i \in Z.

2.5. Route discovery

When source node wants to send a data to a particular destination, at the first it must investigate its cache memory and if has not found a route to destination node, it begins route discovery action.

The first stage of route discovery is started by sending some packets to the adjacent nodes. This work is performed by emission of agents in all directions. These agents are called Forward Agent. When the agents reaches to the destination from various routes, the destination sends a Backward Agent per each of them that via receiving them by source, this source finds that there is a route to destination, and it selects one of the routes as optimized route by considering to the received features from possible routes based on considered parameters. Certainly, other routes also are saved in the cache memory.

2.6. Route Optimality Weight

When a node sends a packet (FA, BA and common packets), each packet carries some value named ROW that indicates optimality value of a route which is transmitting from it. ROW has a value between zero and 1 that whatever it is greater, the possibility of route selection is higher. Its primary value is 1. when a packet reaches to a node, ROW value of that packet is multiplied in NSV (Node Status Value) and the result is replaced as a new ROW in packet. (see Figure 1).

\[
\text{ROW} = \Pi \text{ (NSV of intermediate nodes from source to destination)}.
\]
Coincident with reaching FA to destination, Row values is received by destination and the received same value of ROW is piggyback to the source by applying BA. (according to Figure 1).

2.7. Possibility of Route Selection

Selecting a route among total routes between source and destination is also the under discussion issue. By considering to this matter that routes are arranged basis on ROW, therefore whatever value of ROW is greater, the possibility of route selection is higher. But for multipath routing, we send traffic over n-best routes based on their ROW in a round robin manner.

2.8. Route Maintenance

In MA-DSR Route Maintenance mechanism, On receiving the Update message from the destination, the source updates the cache with the ROW values as shown below;

\[
\text{New ROW Value} = \alpha \times (\text{Old ROW Value}) + (1-\alpha)(\text{Received ROW Value})
\]  

(6)

That the initial value of \( \alpha \) is considered 0.25.

3. MA-DSR Improvement with Route Error Analysis

After study on MA-DSR behavior, simulation results and analysis them, we propose a new cross layer approach and some modifications in the MAC layer. By this way, we can detect reason of RERR reception and apply proper mechanisms for performance improvement. In this section we describe our idea;

In DSR routing protocol, When a sender node attempts to communicate with another neighbor node, the communication may fail. This unsuccessful communication may happen because either the receiver node is unreachable by the sender node (broken link), or there is congestion at the receiver node. However, the routing layer always interprets this failure as broken links and after arriving RERR packets, routes that contain these links eliminated from route cache. But, by this strategy two questions posed in the mind;

Is any RERR as a result of mobility of nodes and routing algorithm must remove routes that contain these nodes from route cache?

Isn’t better that we detect and analyze reason of RERR packets, and don’t eliminate routes blindly?

Accurate answers and solutions for posed questions are the base of our work in this section.

We adapt MA-DSR routing protocol using information from MAC layer, detecting error links and avoiding unnecessary route maintenance processes. Standard DSR, selects routes from cache based on their hop count. In adaptive MA-DSR we can change this strategy by multiplying ROW value in a coefficient, for routes that contain escaped, low power, congested or insecure nodes (pseudo code).

Hence, in our modified algorithm, the received power of each neighbor node at MAC layer was stored away in every node and was used later to inform the routing protocol when a transmission among nodes was unsuccessful. Moreover, we calculate average received signals strength in each node and compare it with transmitted signal strength. The reason of using an average value, instead of only the last received value from neighbour nodes, is to adapt this approach to more realistic scenarios, where objects such as furniture, may interfere temporally in communication among mobile nodes.

By tracking the average signal strength of each node, when DSR protocol launches route error and it’s from weakness of signal strength, two reasons come in our mind;

1. If nodes battery power is adequate, node movement will cause this Error,

2. Otherwise, if that node has battery low problem, node is immobile and battery low problem has limited its transmission range [7].

Here, our algorithm deletes routes that contain this mobile or low power node from cache and apply specific policies for future use of this node in routing algorithm. But, when above problems don’t occur, our algorithm will check possibility of congestion problem. This can be found from queue length. Therefore, we test fullness percentage of queue and use its value in our decision. If this value is higher than threshold (considered in algorithm), congestion occurred. Because existence of many packets in node’s queue can result long latency for packets or even packets
dropping [8]. In this case, algorithm don’t delete route from cache but reduce its selection probability by changing its ROW parameter. Finally, if sender received RERR packet and protocol don’t detect low power, mobility and/or congestion problem, this may be an attack from malicious nodes which do not cooperate with other nodes and try to disrupt the network. We introduce black list for dangerous nodes and change ROW of routes that containing them (see pseudo code).

RERR Analysis:

```c
{ // Mobility and Low battery
  If (average of recent received powers ≤ threshold) {
    event = link breakage; delete route from cache;
    if (node battery power ≤ threshold) set node in red list;
    else act based on rules 1- 4.
    find (or select) new route; }
  //Congestion
  else if (queue being full) {
    event = congestion; calculate new ROW by considering node status based on rules and table 1; select a route;
  }
  //Security
  else {
    event = security problem and may be an attack;
    calculate new ROW by considering node status based on rules and table 1; select a route;
  }
}
```

For better performance we introduce three lists and keep some nodes in them according to below rules. Each list has constant length (Table 1). Rules that implied in new routing mechanism:

Rule 1; We set very escaped or congested nodes in yellow list after 5 times and in red list after 8 times.

Rule 2; If a node declare RERR more than 10 times without any reason, it’s may be malicious node and we set it in black list.

Rule 3; We set nodes that their battery is low (less than 20%) in red list. By this way nodes that their battery lower than other, act as a router seldom.

Rule 4; If routes contain nodes that redound any problem erstwhile, their ROW multiply in 1.25, but if these nodes in one of the three lists, we act based on coefficients in Table 1.

<table>
<thead>
<tr>
<th>List</th>
<th>Size</th>
<th>ROW Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yellow</td>
<td>(Number of Nodes) / 5</td>
<td>1.5</td>
</tr>
<tr>
<td>Red</td>
<td>(Number of Nodes) / 10</td>
<td>1.75</td>
</tr>
<tr>
<td>Black</td>
<td>(Number of Nodes) / 10</td>
<td>2</td>
</tr>
</tbody>
</table>

4. Simulation Results

Extensive simulations were carried out to compare the condition aware MA-DSR routing protocol proposed in this paper with the conventional DSR. The use of network simulations could be understood as a cheaper way of protocol validations (both money and time), where the experimental conditions can be controlled. Network Simulator (NS-2) is used to simulate these protocols.

4.1. Simulation Environment

Two different scenarios are considered in this case, one with 10 nodes in a 500x400m² network and the other with 50 nodes in a 1000x1000m² network. The sources in the network are CBR (constant bit rate) traffic generators. The data packets used for simulations are 1024 bytes. All simulations are carried out for 300 seconds. The MAC layer uses the 802.11b protocol for the wireless standard. The radios use Two Ray Ground propagation model and have a
receiving range of 250 meters. All the nodes in the network are constantly moving and have a random speed selected
in the range of 0-5m/sec.
The key performance metrics evaluated in the simulations are:
- Packet Delivery Ratio; Ratio of number of packets delivered to number of packets sent.
- Routing Overhead; Ratio of number of control packets to number of messages sent.

4.2. Simulation Results

The first set of simulations is performed on a 10-node network in a 500x400m² area. The results are the average of
the performance metrics over 3 simulations with different 10-node scenarios with the same traffic. The traffic load
here are 2 sources sending 1024 bytes data at an interval of 0.1 seconds.

<table>
<thead>
<tr>
<th>Metric</th>
<th>DSR</th>
<th>MA-DSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet Delivery Ratio</td>
<td>99.921</td>
<td>99.915</td>
</tr>
<tr>
<td>Control Packets</td>
<td>27</td>
<td>416</td>
</tr>
<tr>
<td>Message Packets</td>
<td>7563</td>
<td>7987</td>
</tr>
<tr>
<td>Routing Overhead</td>
<td>0.35%</td>
<td>4.9%</td>
</tr>
</tbody>
</table>

From Table 2, we see that the performance of both Multi Agents based Algorithm and DSR are similar in terms of
packet delivery ratio and average hops. However the Multi agent based algorithm has higher overhead due to the
continuous exchange of Backward Agents in the route maintenance phase for updating the goodness of a particular
route. However the biggest difference is in load balancing and energy management. Figure 2 shows the number of
packets sent/forwarded by each node in the network. The DSR results show a chaotic distribution of packets
amongst the nodes.

Some of the nodes (nodes 9 & 10) forward an extremely high number of packets results in high energy utilization
while others are relatively idle. However in the MA-DSR Algorithm, since multiple routes are used to send packets,
the loads are even distributed amongst most of the nodes. Average end to end delay test in 10-node scenario, shows
MA-DSR has better performance than Standard DSR(see Figure 2).

The second set of simulations is performed on a 50-node network in 1000x1000m² field area. Also 3 CBR traffic
sources are used which send packets of size 1024 bytes at an interval of 0.1 seconds.

From Table 3, we see that although the MA-DSR has a higher routing overhead than DSR, it has a better
performance in terms of packet delivery ratio. Also from Figure 3, we see that the load distribution is uneven in
DSR with only 10% of the nodes involved in active traffic.

Also some nodes handle heavy traffic forwarding (more than 15000 packets). However in MA-DSR Algorithm,
more nodes (about 30%) are involved in active traffic forwarding and also the traffic is more distributed amongst the
nodes. Thus MA-DSR Algorithm provides better load and energy balancing than DSR.
By more tests, we see that in large scale networks MA-DSR shows better performances too. For example, by calculating average end-to-end delay for 50-node scenario, MA-DSR is better than Standard DSR (see Figure 3).

![Figure 3. (a) Load Distribution; (b) Average end-to-end delay in 50-node scenario.](image)

### 5. Conclusions and Future Works

Mobile Ad-hoc Networks provide a wide array of challenges in routing and network management. In this paper we present a novel optimization for DSR routing protocol based on multi agent. We acquire some properties such as signal strength, shortest-path metrics, congestion measurement and energy level in each node, then use them in route selection decision. Then our protocol uses a combination of information such as nodes mobility, battery power, congestion and security status for analysis link failures. By using these information we determine the cause of route breakage and apply changes in the routing decision policies of MA-DSR protocol. This algorithm has a lot of scope for future improvements. In perspective, we shall adapt the movement of agents to the mobility. In this sense, they should not circulate with the same rate in a zone of high mobility as in the zone of less mobility. Intelligent broadcasting for broadcast storm prevention and eliminating some preconditions about equal transmission range of nodes are another aspect for proposed MA-DSR improvement.

### References