A DISCOVERY METHOD FOR NODE-DISJOINT MULTI-PATHS BY VALID SOURCE-DESTINATION EDGES

In Ad hoc networks most on demand routing protocols concentrate on the single path from a source to a destination. Exhaustively using a single path increases traffic load and faster consumes energy of nodes on the path. Therefore, preventing concentration on a single path and distribution of energy and traffic load on whole network is an important issue on Ad hoc networks. Some searching methods for multi-paths are studied, but finding all available paths from a source to a destination is a very hard studying topic.

In this study, we provide an efficient and a straightforward node-disjoint multi-path discovery algorithm for searching all available paths. The routing method is provided based on AODV. We implement simulation models for analyzing efficiency of proposed multi-path-finding method on the NS-2. Simulation results show that the proposed methods can get very good results on multi-path finding rate.

1. INTRODUCTION

A Mobile ad hoc network is a collection of mobile nodes that can communicate with each other using multi-hop wireless links without utilizing any based station infrastructure and centralized management. Each node in the network acts as both a host and a router. The design of an efficient and reliable routing protocol in such a network is a challenging issue.

On-demand routing protocols in particular, are more preferred because they consume much less bandwidth than table-driven protocols. Ad hoc On-Demand Distance Vector (AODV) and Dynamic Source Routing (DSR) are the two most widely studied on-demand ad hoc routing protocols. Previous works have shown limitations of the two protocols. The main reason is that both of them build and exhaustively use single path route for each data session. Whenever communication link breaks on the active route, each protocol has to invoke a route discovery process. Route discovery flood is associated with significant latency and overhead. Besides, exhaustively using a single path causes for nodes faster consume energy and high traffic load on the path. Therefore preventing concentration on a single path and distribution of energy and traffic load on whole network is a new challenge.

On-demand multi-path routing protocols can alleviate these problems by establishing multiple paths between a source and a destination in a single route discovery. A new route discovery is invoked only when all of its routing paths fail or when there only remains a single path available. The main focus is not choosing multi-path or single path, but how to discover maximum possible complete node-disjoint paths. Discovery of complete node-disjoint multipath has been proposed in [3, 5, 6 and 8]. Although these protocols build multiple paths on demand, most of them could not guarantee to find all available complete disjoint paths even though they use
much complex methods.

In this paper, we propose a novel method of discovering complete node-disjoint multiple paths maximum available in the network. The method modifies and extends AODV to enable path discovery and accumulation feature.

2. DISJOINT PATHS

The term ‘disjoint path’ is used to define how many nodes participate more than twice in discovery of multiple paths from a source to a destination. Several studies proposed various types of disjoint paths, such as [3], [6] and [8]. Generally multipath can be classified in three categories (Figure 1): node-disjoint (complete disjoint), link disjoint, non-disjoint.

- **Node-disjoint** – Each node participates in only and only one path.
- **Link-disjoint** – Nodes may participate in several paths, but not two node or more nodes sequentially.
- **Non-disjoint** – Nodes may participate in several paths and sequentially.

![Figure 1](image.png)

Complete node-disjoint multi-path is much robust than others, this was proven in [3] using path-break probability. Therefore, discovery of node-disjoint multi-path is an important part of multipath routing. Consequently, discovery of all available node-disjoin paths from a source to a destination could be more efficient for many protocols.

Most of on-demand multipath routing protocols such as [5] and [8] use a simple method of multi-path discovery. In this method, source initiates route request whenever needs to send data and destination node sends reply for each received route request even it is from same source. In this way source node simple may build multi-path. However, in several cases multi-path discovery fails and discover much fewer number of paths than currently available. Figure 2 shows one example for those cases.
Figure 2. Multipath discovery failure

When source node broadcast route request (RREQ) node 1 and 2 receives first and pass on next nodes. Some nodes like 4, 5, 6, and 7 may receive redundant RREQ and they discard it (shown in figure as dashed arrow). When destination node receives RREQ from node 6 and 7, the node sends RREP through reverse path. First RREP arrives to source node through D → 6 → 3 → 1 → S and second RREP is discarded on node 1, because node 1 already received RREP. If node 1 does not discard RREP source node could build complete node-disjoint multipath. As result, source node can build only a single path S → 1 → 3 → 6 → D, even though there are other paths.

Another method of multipath discovery is source routing. Protocols such as [3] and [4] use this method. The method discovers multi-paths on demand recording router address in RREQ and destination node decides to select paths. Discovery of multipath using this method may seem very simple, but it causes a lot of overhead and routing load due to heavy RREQ. Inserting address of nodes increases packet size each time and mobile devices currently are not capable of handling heavy packets. This method is not even preferred in wired network with higher capability devices.

3. A DISCOVERY METHOD FOR NODE-DISJOINT MULTI-PATHS

In this section we present an overview and route discovery of proposed method, called a discovery method by valid source-destination edges (VSDE). VSDE discovers available multi-paths from a source to a destination.

3.1 Method Overview

Goal of proposed method is to discover maximum available paths from a source to a destination. Analyzing previous works, we become known that number of multi-paths between source and destination depends mostly on two parameters: edges of source node and edges of destination node. Number of node-disjoint multi-paths is not more than number of edges of source and destination.
As shown in figure 3, even though there are enough intermediate nodes to build more multi-paths, complete node-disjoint multi-paths can be only 3 – edges of source node. Therefore deciding multi-paths according to edges of source and destination would be exact and efficient.

3.2 Path Discovery

Since the discovery method is reactive routing protocol, no permanent routes is stored in nodes. Node-disjoint multipath searching algorithm is shown at figure 5. The source node initiates route discovery procedure by broadcasting the ROUTE REQUEST (RREQ) message. When neighbor nodes of source (one-hop nodes) receive RREQ, the nodes, we call them token nodes, include their address to the message. We also call these nodes secondary sources.

![Figure 3. Maximum multi-paths](image)

All intermediate nodes uniquely identify redundant messages using source address and sequence number. Sequence number is set in source node and incremented when source node initiates route request. Neighbor nodes of destination allow two duplicate messages and if token node (secondary source) is different.

Destination node generates ROUTE REPLY (RREP) for each token node (secondary source).
(1) ALL NEIGHBOR NODES OF A SOURCE, TOKEN NODES, BUILD A SINGLE PATH TO A DESTINATION NODE USING AODV METHOD

(2) NEIGHBOR NODES OF DESTINATION NODE ALLOW UP TO TWO PATHS FROM DIFFERENT TOKEN NODES.

(3) EACH PATH FROM THE EDGE NEIGHBORS OF DESTINATION:
   A. If two token paths share a path, only one token path is selected.
   B. When two paths overlap with same token node, first one will be selected;
   C. If edge neighbor nodes of destination are fewer than token nodes, number of selected paths is not more than number of destination edges;
   D. If all available paths are fewer than edges of source and/or destination due to intermediate nodes, number of selected paths can be less than source edges or destination edges

**Figure 5. Algorithm of multi-path discovery**

Each route reply builds a single path to each token node and source node can build multi-paths with it. If edges of destination node are fewer than edges of source, number of multi-paths is equal to edges of destination. Goal of our method is to build maximum complete node-disjoint multi-paths. If intermediate nodes receive RREP several times due to token address, the nodes just discard it.

After discovery of multi-paths, they can be used for several ways. Hop one by one or use exhaustively. Hopping has different advantages mentioned in [2] such as security, traffic load and energy distribution. Exhaustively using may increase packet delivery ratio and decrease packet overhead.

4. PERFORMANCE RESULTS

We first describe the simulation environment used in our study and then discuss the results in detail.

4.1 Simulation Environment

Our simulations are implemented in Network Simulator (NS-2) [8] from Lawrence Berkeley National Laboratory (LBNL). The simulation parameters are as follows:

- Number of nodes: 20, 30, 40, 50, 75 and 100 respectively;
- Testing area: 1000m x 1000m;
- Mobile speed: 2, 5, 7, 10 and 15 respectively;
- Mobility model: random way point model (when the node reaches its destination, it pauses for several seconds, e.g., 1s, then randomly chooses another destination point within the field, with a randomly selected constant velocity);
- Traffic load: UDP, CBR traffic generator;
- Radio transmission range: 250 m and
- MAC layer: IEEE 802.11.

Each simulation is run for 500 seconds and each case has 10 different topologies.
4.2 Path Discovery Performance

To evaluate proposed VSDE path discovery performance we compare it with theoretical results. To compare path discovery performance we use a pair of nodes as the source and the destination. Nodes are fixed in a position and do not move. Figure 6 shows discovered number of paths according to the total number of nodes, i.e. density. AVG means average value of 10 different topologies and MAX means maximum discovered paths. VSDE average and maximum are same with actual results (ACT) when number of nodes is less than 50, because a few available multi-paths exist in the network when density is low. An actual result means that the number of all available paths that is counted manually.

![Figure 6. Path Discovery](image)

![Figure 7. Path Discovery Rate](image)

However, when network density is high, more actual paths are discovered. We can see VSDE discovery rate, which is ratio of discovered paths by VSDE to actual method, in figure 7.

4.2 Routing Performance

We evaluate routing performance of VSDE using exhaustively path utilization by FCFS method and compare it with AODV using three metrics.

- Delivery Rate: the ratio of packets reaching to the destination node comparing to the total packets generated at the source node.
- Average Energy Remained: mean value of energy remained in each node.
- Control Overhead: sum of all route request messages, route reply messages and route error messages.

Figure 8 and 9 show packet deliver ratio of each protocol when number of nodes and node speed varies. When there are more nodes in network VSDE can show better performance because more multi-paths are available. When node speed increases path break probability also increases, therefore VSDE packet delivery ratio become slightly less than AODV.
Packet Delivery Ratio
Node speed 2m/s, 3 sources

<table>
<thead>
<tr>
<th>Number of nodes</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>75</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>84.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>86.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>88.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>90.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>92.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>94.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>96.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>98.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Number of nodes

Figure 8. Packet Delivery Ratio, when the number of nodes varies

Figure 9. Packet Delivery Ratio, when the node speed varies

Average Remained Energy
Node speed 2m/s, 3 sources

<table>
<thead>
<tr>
<th>Number of nodes</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>75</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Number of nodes

Figure 10. Average energy remained, when the number of nodes varies

Figure 11. Average energy remained, when the node speed varies

Figure 10 and 11 shows the average remained energy of each protocol. We have to mention that it is a mean value of remained energy of each node at the end of simulation. VSDE maintains lower energy consumption because it floods less route requests.

Packet Overhead
Node speed 2m/s, 3 sources

<table>
<thead>
<tr>
<th>Number of nodes</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>75</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Number of nodes

Figure 12. Control Packet Overhead, when the number of nodes varies

Figure 13. Control Packet Overhead, when the node speed varies
Figure 12 and 13 shows the control packet overhead of each protocol according to the variation of speed and number of nodes. Packet overhead of each protocol increases in proportion to the increase of node density and speed. Because more requests are flooded over network, yet VSDE shows better performance in it.

5. CONCLUSION

This paper presents a method for finding maximum node-disjoint paths between a pair of nodes in an on-demand manner. We transform the problem of finding maximum node-disjoint paths to the problem of finding complete disjoint paths from token nodes to destination. We propose an algorithm of token-node method for discovery of maximum node-disjoint paths in an ad hoc network. We also present an example protocol that extends AODV to find node-disjoint paths. Simulation results show that our method guarantee over 80% discovery of available node-disjoint paths on-demand in wireless ad hoc networks.

REFERENCES


