RZRP: A Pure Reactive Zone-based Routing Protocol with Location-based Predictive Caching Scheme for Wireless Mobile Ad Hoc Networks

Weibo Chen, Kun Yang and Achilleas Achilleos
Department of Computing and Electronic Systems
University of Essex
Colchester, CO4 3SQ United Kingdom
Email: {wchen, kunyang, aachila}@essex.ac.uk

Abstract— Hybrid routing strategy is widely utilised in hierarchical routing protocols to balance the control overheads and packet delivery delay in mobile ad hoc networks (MANETs). In these protocols the area of the concerned MANET is usually divided into zones. As not all of the zones have the equal probability to become an active relay zone, the resources for proactive route information maintenance in each zone are then wasted. Such waste can be significant in large-scale networks. To cope with this deficiency we propose a two-stage pure reactive solution called RZRP. However, the pure reactive implementation can lead to longer delay than proactive or hybrid protocols. To solve this problem, a location-based predictive caching scheme is designed to reduce the latency caused by reactive route discovery. The simulation results have shown that the performance of the RZRP protocol has significantly improved in terms of reducing communication overhead and packet transmission delay.

I. Introduction

Due to node mobility and limited network resources, one of the key challenges in MANET routing design is how to maximize the success of data packet delivery at as low cost of network resources as possible under rapid change of network topology. Recent researches have showed that hierarchical routing structure can efficiently improve the performance of routing protocols in terms of scalability and robustness [1-5]. The protocols presented in [1] and [2] partition the whole network into fixed non-overlapped small geometrical areas based on location coordinates. Whereas in [3-5], the network is partitioned based on nodal connectivity information, which pre-defines a zone radius in hops. Any node whose distance in hops to the central node is less than or equal to the zone radius will be treated as local neighbours of the central node. Therefore the node is in the routing zone of the central node, and any node that requires packet transmission must form its own routing zone. Hence the routing zone may be created dynamically and they overlap with each other.

Moreover, some of hierarchical protocols require the existence of gateway node or cluster head in each zone for central administration and packet relay. The cluster head concept can improve routing performance but may result in fast power depletion of head nodes. Furthermore, extra communication cost is also required for head election and cluster structure maintenance.

Whatever approach being used, one common aspect that can be found amongst the above protocols is the employment of hybrid routing strategy, i.e. a combination of proactive and reactive solutions. By using hybrid routing strategy, the local or global topological information is maintained proactively, whereas route discovery packet is initiated reactively. As in a large-scale network, not all zones have the equal probability to become an active relay zone. The network resources such as bandwidth and energy may be greatly wasted in these less-frequently-used zones. In this paper, we propose a pure reactive two-level routing approach called RZRP for zone-based routing protocol in order to make best use of overall network resources such as bandwidth and energy while fulfilling routing tasks. As purely reactive implementation may result in longer packet delivery delay, a location-based predictive caching scheme is then incorporated into RZRP to remedy this drawback.

The rest of this paper is organized as follows. Preliminary knowledge is presented in Section 2. Section 3 describes the operations of this protocol. The mathematic analysis and evaluation results are presented in Section 4 and 5 respectively. Finally, this paper concludes at Section 6.

II. Preliminaries

This section presents the assumptions, data structures and a location-based link expiry time prediction method of RZRP.

A. Assumptions

Due to the utilization of location information, all nodes in RZRP are assumed to be equipped with GPS receiver or equivalent equipment to get information like geographic location coordinates, current time, moving speed and direction. Link between two nodes is assumed to be symmetric, and a uniform velocity linear movement model is adopted for each node during the period from current time until the time when the link broken. The network partition method is similar to that in [1] and we also assume that all nodes in the network already know the partition information such as zone ID and scope of each zone via some simple calculation if given the side lengths of zones.

B. Data Structures

The major control packets in RZRP are Inter-zone RREQ Packet, Intra-zone RREQ Packet, Inter-zone RREP Packet, and Intra-zone RREP Packet. Their structures are showed as follow:

Inter-zone RREQ: <RREQ_ID, SourceNode_ID, SourceZone_ID, DestNode_ID, rZone_List> where the RREQ_ID and SourceNode_ID are utilized to identify a packet. The rZone_List contains the IDs of zones that this packet has passed through.

Intra-zone RREQ: <RREQ_ID, InitNode_ID, InitZone_ID, DestNode_ID, LastHop_loc, Route_List> this packet triggers intra-zone route discovery. The LastHop_loc field contains location information of previous node, which will be used for the location-based predictive caching.

Inter-zone RREP: <RREP_ID, rplyNode_ID, rplyZone_ID, Routes, Expiry> where the Routes field contains complete path from source zone to destination zone.
in a zone-to-zone manner (i.e. the route is composed of only zone IDs). The Expiry field contains a time indicating when a routing path becomes invalid.

Intra-zone RREP: \(<\text{RREP}\_\text{ID}, \text{rplyNode}\_\text{ID}, \text{rplyZone}\_\text{ID}, \text{Links}\_\text{List}\>\) where the \text{Links}\_\text{List} field contains individual links which are 1-hop connections in a node-to-node manner.

To implement the location-based predictive caching scheme, all nodes are required to maintain three tables: Inter-link table, Intra-link table, and Path table.

### Table I: Notation Used in Prediction

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n_i, n_j)</td>
<td>The nodes at each end of a link.</td>
</tr>
<tr>
<td>(l_{ij})</td>
<td>The link connects (n_i) and (n_j).</td>
</tr>
<tr>
<td>(r)</td>
<td>The maximum transmission range of node</td>
</tr>
<tr>
<td>(d_{ij})</td>
<td>The current distance between (n_i) and (n_j)</td>
</tr>
<tr>
<td>(v_i)</td>
<td>The speed of (n_i)</td>
</tr>
<tr>
<td>(\theta_i)</td>
<td>The moving direction of (n_i)</td>
</tr>
<tr>
<td>((x_i, y_i))</td>
<td>The location coordinates of (n_i)</td>
</tr>
<tr>
<td>(t_{\text{current}})</td>
<td>The current time</td>
</tr>
<tr>
<td>(t_{\text{break}})</td>
<td>The period from (t_{\text{current}}) until (l_{ij}) broken</td>
</tr>
</tbody>
</table>

In RZRP, a caching scheme is implemented to enhance the performance of RZRP to improve route reliability and to reduce route discovery/recovery cost. This caching scheme utilizes a prediction on the status of links based on the movement information of both ends of links. The transmission of such information over network will increase the network resources usage. To minimize it we piggyback this information to the intra-zone RREQ packet, and such information is only allowed to be transmitted to 1-hop away neighbours. Once the node knows the movement information of its previous node, the time by which the distance between them reaches the maximum transmission range can be calculated.

According to [14], the value of connection breaking points \((x_j', y_j')\) and \((x_j, y_j)\) can be calculated by the following formulas:

\[
x_j = x_j - v_j \cdot t_{\text{break}} \cdot \cos \theta_j \quad (1)
\]
\[
y_j = y_j - v_j \cdot t_{\text{break}} \cdot \sin \theta_j \quad (2)
\]
\[
x_j' = x_j - v_j \cdot t_{\text{break}} \cdot \cos \theta_j \quad (3)
\]
\[
y_j' = y_j - v_j \cdot t_{\text{break}} \cdot \sin \theta_j \quad (4)
\]

As shown in Figure 1, the distance \(d\) between \(n_j\) and \(n_j\) can be calculated by using the Pythagorean Theorem:

\[
d^2 = (x_j' - x_j)^2 + (y_j' - y_j)^2 \quad (5)
\]

As the link \(l_{ij}\) breaks when \(d \geq r\) we have

\[
r^2 \geq (x_j' - x_j)^2 + (y_j' - y_j)^2 \quad (6)
\]

By applying (1), (2), (3), and (4) to (6), the equation can be rewrte as

\[
r^2 \geq [(x_i - v_i \cdot t_{\text{break}} \cdot \cos \theta_i) - (x_j - v_j \cdot t_{\text{break}} \cdot \cos \theta_j)]^2 + [(y_i - v_i \cdot t_{\text{break}} \cdot \sin \theta_i) - (y_j - v_j \cdot t_{\text{break}} \cdot \sin \theta_j)]^2 \quad (7)
\]

This also equals to

\[
r^2 \geq [(x_i - x_j) - (v_i \cdot \cos \theta_i - v_j \cdot \cos \theta_j) \cdot t_{\text{break}})^2 + [(y_i - y_j) - (v_i \cdot \sin \theta_i - v_j \cdot \sin \theta_j) \cdot t_{\text{break}})^2 \quad (8)
\]

Let \(a = x_i - x_j\), \(b = v_i \cdot \cos \theta_i - v_j \cdot \cos \theta_j\), \(c = y_i - y_j\), and \(d = v_i \cdot \sin \theta_i - v_j \cdot \sin \theta_j\) the equation (8) can be reformd as

\[
r^2 \geq (a - b \cdot t_{\text{break}})^2 + (c - d \cdot t_{\text{break}})^2 \quad (9)
\]

By transforming equation (9) to quadratic format we have

\[
(b^2 + d^2) \cdot t_{\text{break}}^2 - 2(ab + cd) \cdot t_{\text{break}} + (a^2 + c^2 - r^2) \geq 0 \quad (10)
\]

Therefore, the value of \(t_{\text{break}}\) can be calculated by using the quadratic formula

\[
t_{\text{break}} \leq \frac{(ab + cd)}{b^2 + d^2} + \sqrt{(ab + cd)^2 + (b^2 + d^2)(a^2 + c^2 - r^2)} \quad (11)
\]

Hence, \(t_{\text{current}} + t_{\text{break}}\) is identified as the expiry time of \(l_{ij}\).
III. Operation of the Proposed Protocol

A. Overview of Protocol Operations

The essence of this protocol lies in the integration of pure reactive route discovery and location-based predictive caching scheme. The pure reactive implementation is carried out via two-stage route discovery: inter-zone route discovery and intra-zone route discovery. The purposes of inter-zone route discovery are two-fold. Firstly, it establishes the inter-zone route between the source zone and destination zone in a zone-to-zone manner. Secondly, it triggers intra-zone route discovery if there is no valid routes to neighbouring zones. Intra-zone route discovery has the responsibility of confirming existence of the destination node in a zone, discovering the connectivity status of nodes inside the zone and the connectivity status with neighbouring zones. Moreover, as the location information is piggybacked to intra-zone RREQ packet, the intra-zone route discovery also triggers link’s expiry prediction at each node. The source node using the source routing strategy decides a complete inter-zone routing path from source zone to the destination zone in zone-to-zone manner. The intermediate nodes inside each zone that along the inter-zone routing path decide which neighbouring node can be used to forward the data packet towards next routing zone.

As described early, a caching scheme is implemented to improve the performance of RZRP in terms of reducing the total number of route discovery requests and route establishing latency. However, the issue of this implementation is that the “freshness” of cached entries must be guaranteed. As these entries either be removed too early or too late will result in severe performance degradation. To solve this problem, we use the location-based prediction method to predict the link’s expiry time, and the result will be used as TTL of that entry in cache. Compare to the packet-based route information update method, this implementation requires less network resources. As all necessary information is piggybacked to route discovery packets the cost of cache maintenance is then minimized.

B. Two-stage Reactive Route Discovery

Figure 2 shows an example of transmitting an inter-zone RREQ packet from source node S to destination node D. The inter-zone RREQ packet is initiated at S when the node ID of D cannot be found either in the intra-link table or path table of S. After the initiation S is then search the path table for valid paths to its neighbouring zones. If it cannot find any valid path to its neighbouring zone an inter-zone route discovery will be initiated and propagated inside the local zone of S. Otherwise, S forwards the inter-zone RREQ packet following the existing internal path to the node b.

![Figure 2: an example of inter-zone route discovery](image)

As Procedure 1 showed, on receiving the inter-zone RREQ packet from neighbouring zone, b firstly has to make sure that the packet is never received before. After that b checks its intra-link table. An inter-zone RREP packet will be initiated and sent back to source node only if the table contains the destination node ID or b is the destination node. Otherwise, the path table will be checked. If b is the first node in the zone received this packet and cannot find any internal paths to the neighbouring zones, an intra-zone route discovery will be initiated. This process repeats until the packet reaches destination node.

### Procedure 1: Processing inter-zone RREQ

1. begin
2. if received this packet before
3. drop this packet
4. else if can find the destNode_ID in intra-link table or my.Node_ID == destNode_ID
5. send inter-zone RREP packet back
6. else if can find valid paths to neighbouring zones in path table
7. forward this packet
8. else if the packet contains my.Zone_ID == false
9. initiate intra-zone route discovery
10. add my.Zone_ID to the packet
11. end

### Procedure 2: Processing inter-zone RREP

As Procedure 2 indicates, on receiving an inter-zone RREP packet, the intermediate nodes have the responsibility to update the Expiry filed of the packet and only the destination node that assigned in the packet can cache the path directly to its path table.

![Figure 3: an example of intra-zone route discovery](image)

As Figure 3 shows, node h and b initiate their own intra-zone RREQ packet on receiving the same inter-zone RREQ packet from node a. As these two packets share the same packet ID, one of them will be discarded silently when arriving at node f since f treats them as the same. Therefore, the risk of triggering multiple intra-zone route discoveries by a single inter-zone RREQ packet is avoided.

1. begin
2. get the movement information from packet
3. predict the link’s expiry time
4. if received this packet before
update my.intra-link_table 
6. drop this packet
7. else if my.Zone_ID != packet.initZone_ID 
8. generate an intra-zone RREP 
9. send the RREP back to packet.initiator 
10. cache the link to my.inter-link_table 
11. else if my.intra-link_table contains packet.destNode_ID 
12. overwrite the movement information 
13. forward this packet to my neighbours 
14. cache the link and the links in packet.Route_List to my.intra-link_table 
15. send inter-zone RREP back to source node 
16. else 
17. overwrite the movement information 
18. forward this packet to my neighbours 
19. cache the link and the links in packet.Route_List to my.intra-link_table 
20.end 

Procedure 3: Processing Intra-zone RREQ

As we can see from Procedure 3, when f receives the intra-zone RREQ packets from b and h, the first thing is getting the location information and predicting the expiry time of the connections between them. If f finds these two intra-zone RREQ packets have the same packet ID, the later one will be discarded silently. The link information carried by the late packet will be put into the intra-link table in order to construct the local network structure. If the intra-link table already cached this information, the expiry time of the entry will be updated with the new predicted time. This process repeats until the packet reaches a node at outside of the initiator’s local zone.

1. begin 
2. if already received this packet from the previous node 
3. drop this packet 
4. else 
5. cache packet.Links.List.inter-link to my.intra-link_table 
6. cache packet.Links.List.intra-links to my.intra-link_table 
7. forward this packet 
8. generate paths to the initiator of this packet 
9.end 

Procedure 4: Processing Intra-zone RREP

The process of intra-zone RREP packet is following Procedure 4. After forwarding an intra-zone RREP packet, the node is then required to construct paths to that neighbouring zone based on the information from the intra-zone RREP packet. These paths will be stored in the path table.

C. Route Selection

| TABLE II: NOTATION USED IN SELECTION |
|-----------------|-----------------|
| Notation        | Definition      |
| d               | The neighbouring zone of current zone |
| D               | The destination zone |
| S_{int ra}      | An intra-path set to d |
| S_{int er}      | An inter-path set to D |
| R_{int ra}      | The selected route to d |
| R_{int er}      | The selected route to D |
| t_{int ra}      | The lifetime of R_{int ra} |
| T_{int er}      | The lifetime of R_{int er} |
| l_{int ra}^{i}  | The i^{th} link in S_{int ra} |
| l_{int er}^{i}  | The i^{th} link in S_{int er} |
| N_{int ra}^{i}  | The number of hops of l_{int ra}^{i} |
| N_{int er}^{i}  | The number of hops of l_{int er}^{i} |
| t_{p}           | The lifetime of the path |

As the source node predefines routing path to the destination node, it may have to make a selection when the path table contains more than one path to that destination. The following algorithms are implemented to help the source node and intermediate node make decisions. The primary consideration of selecting an intra-path is the number of hops of that path. This is because of that zone level connection is more robust than node level connection. By selecting an inter-path with less number of hops means that the packet relay involves less number of nodes. Differing from Algorithm 2, the primary consideration of selecting an intra-path is the lifetime of that path. By using a path with longest lifetime implies that the number of route discovery requests can be minimized.

Intra-path Selection Algorithm

input: S_{int er} - the inter-paths set to D 
output: \{ R_{int ra}^{i} , T_{int ra}^{i} \} - the selected intra-path to d and the lifetime of the path 

begin 
1. set T_{int ra} = 0, R_{int ra} = 0 
2. for \( p_{int ra}^{i} \in S_{int ra} \), \( i = 1, \ldots, |S_{int ra}| \) do \{ 
3. set t_p = 0 
4. for \( l^{i} \in p_{int ra}^{i} \), \( i = 1, \ldots, |p_{int ra}^{i}| \) do \{ 
5. if \( t_p == 0 \) then set \( t_p = t^{i} \) 
6. else if \( t^{i} < t_p \) then set \( t_p = t^{i} \) 
7. if \( T_{int ra} == 0 \) and \( R_{int ra} == 0 \) then set \( T_{int ra} = t_p \), \( R_{int ra} = p_{int ra}^{i} \) 
8. else if \( T_{int ra} = t_p \) and \( N_{int ra}^{i} < N_{int ra} \) then set \( R_{int ra} = p_{int ra}^{i} \) 
9. else if \( T_{int ra} < t_p \) then set \( T_{int ra} = t_p \), \( R_{int ra} = p_{int ra}^{i} \) 
10. return \( T_{int ra} , R_{int ra} \) 
end 

Inter-path Selection Algorithm

input: S_{int er} - the inter-path set to D 
output: R_{int er} - the selected inter-path to D 

begin 
1. set R_{int er} = 0 
2. for \( P_{int er}^{i} \in S_{int er} \), \( i = 1, \ldots, |S_{int er}| \) do \{ 
3. if \( R_{int er} == 0 \) then set \( R_{int er} = P_{int er}^{i} \) 
4. else if \( N_{int er}^{i} < N_{int er} \) then set \( R_{int er} = P_{int er}^{i} \) 
5. else if \( T_{int er} = t_p \) and \( T_{int er} < t_p \) then set \( R_{int er} = P_{int er}^{i} \) 
6. return \( R_{int er} \) 
end 

Algorithm 1: Intra-path Selection Algorithm

Algorithm 2: Inter-path Selection Algorithm
IV. Mathematical Analysis

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>The total number of nodes</td>
</tr>
<tr>
<td>l</td>
<td>The total number of links</td>
</tr>
<tr>
<td>z</td>
<td>The total number of zones</td>
</tr>
<tr>
<td>int ra</td>
<td>Average length of intra-path</td>
</tr>
<tr>
<td>int er</td>
<td>Average length of inter-path</td>
</tr>
<tr>
<td>k</td>
<td>Transmission request rate per second</td>
</tr>
<tr>
<td>c</td>
<td>The probability of creating routes for a transmission</td>
</tr>
<tr>
<td>t proc</td>
<td>Packet process delay</td>
</tr>
<tr>
<td>t prop</td>
<td>Packet propagation delay</td>
</tr>
<tr>
<td>b</td>
<td>Broadcast interval</td>
</tr>
<tr>
<td>T</td>
<td>Simulation time in seconds</td>
</tr>
</tbody>
</table>

Generally speaking, the packet delivery delay in a routing protocol is caused by the route creation delay and packet propagation delay. Therefore, in RZRP, the packet delivery delay can be calculated by the following formula:

\[ D_{RZRP} = Q \times \left( d_{\text{int ra}} + \sum_{i=1}^{i=n} d_{\text{int er} \times \text{int ra}} + \sum_{m=1}^{m=\text{int ra}} d_{m} \right) \]  (12)

Where \( Q = k \times T \times c \) is the total number of route creation requests during simulation time \( T \). \( d_{ij} = t_{\text{proc}} \times \frac{n}{z} \) is the delay caused by processing of inter-zone route discovery packet at a zone. \( d_{ij} = t_{\text{proc}} + t_{\text{prop}} \times \frac{n}{z} \) is the delay caused by reactive searching of destination zone ID at each zone. \( d_{m} = t_{\text{prop}} \) is the propagation delay of data packet at each intermediate node.

For hybrid zone-based protocols, such as ZHLS [1], due to the proactive maintenance of route information inside each zone, the packet delivery delay should be shorter than RZRP, which can be represented as:

\[ D_{ZHLS} = Q \times \left( d_{i} + \sum_{i=1}^{i=n} d_{i} + \sum_{m=1}^{m=\text{int ra}} d_{m} \right) \]  (13)

Where \( Q = k \times T \times c \) is the total number of route creation requests during simulation time \( T \). \( d_{i} = t_{\text{proc}} + t_{\text{prop}} \times \frac{n}{z} \) is the delay caused by reactive searching of destination zone ID at each zone. \( d_{m} = t_{\text{prop}} \) is the propagation delay of data packet at each relay node.

As a resource constrained network, the total number of control packets transmitted over the network is another important metric for ad hoc routing performance observation. Thanks to the two-stage reactive route discovery, the total number of control packets can be efficiently reduced in RZRP. Following formula indicates the total number of control packets propagated over the network for route creation:

\[ H_{RZRP} = Q \times \left( \sum_{i=1}^{i=n} (P_{i}^{\text{int er}} + P_{i}^{\text{int ra}}) \right) \]  (14)

Where \( Q = k \times T \times c \) is the total number of route creation requests during simulation time \( T \). \( P_{i}^{\text{int er}} = r_{\text{int ra}} \) is the total number of inter-zone route discovery packet propagated in a zone. \( P_{i}^{\text{int ra}} = \frac{n}{z} \) is the total number of intra-zone route discovery packets propagated in a zone.

In ZHLS, the total number of control packets propagation for route creation can be calculated by the following formula:

\[ H_{ZHLS} = Q \times \left( \sum_{i=1}^{i=n} \frac{n}{z} \right) \]  (15)

Where \( Q = k \times T \) is the total number of route creation requests during simulation time \( T \). \( q_{i} = r_{\text{int ra}} \) is the total number of destination zone ID searching packets propagated in a zone. \( T = \frac{1}{b} \) is the number of broadcasting times during \( T \).

\[ P_{j}^{\text{NLSP}} = \left( n + z \right)^{2} \] is the total number of node link state packets propagated in a zone. \( P_{j}^{\text{ZLSP}} = n \) is the total number of zone level state packets propagated in a zone.

V. Evaluation Results

The benchmark is ZHLS [1], as both ZHLS and RZRP use the same network partitioning approach. ZHLS is a hybrid hierarchical routing protocol. RZRP is implemented in two versions: with cache and without cache in order to observe the impact of caching scheme.

Figure 4 shows the packet delivery delay of both RZRP and ZHLS along the total number of nodes. The increasing of total number of nodes in network implies that both the node density and frequency of route discovery in each zone increase. As Figure 4 shows, the implementation of RZRP without cache suffers longer delay than ZHLS. This is understandable in that reactive protocols usually have longer end-to-end delay than proactive protocol, as the routing path to destination node and neigbouring zones are created on-demand rather than pre-decided on a periodic basis. However, by implementing the location-based predictive caching mechanism in RZRP, such delay can be sharply reduced. Due to the reduction of the number of route discovery requests, the packet delivery delay of RZRPC is even shorter than ZHLS. As the average number of nodes in each zone increases, the increase of packet delivery delay of these three implementations is visible.
VI. Conclusions

Hierarchical routing structure improves the performance of routing protocols in terms of scalability and robustness. However, in order to balance the control overhead generated by the routing protocols and the data packet delivery delay caused by route discovery, hybrid routing strategy is widely utilized in hierarchical routing structure. As MANET is a resource constrained network, network resources are consumed unnecessarily in these zones that may not become an active relay zone. Therefore, in this paper we present a pure reactive zone-based two-level routing protocol with a location-based predictive caching scheme for MANETs. Through the evaluation, our protocol reduces both the control overhead and packet delivery delay at the same time via the combination of two-stage reactive route discovery and location-based expiry time prediction caching mechanism. In order to reduce more control packets over the network, our future development is to investigate an efficient broadcast approach for route discovery packets propagation.

References