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

978-1-4244-2202-9/08/$25.00 © 2008 IEEE

in a zone-to-zone manner (i.e. the route is composed of only zone IDs). The Expiry filed contains a time indicating when a routing path becomes invalid.

Intra-zone RREP: <RREP_ID, rplyNode_ID, rplyZone_ID, Links_List> where the Links_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.

Inter-link Table

LocalGate	NeighGate	NeighZone	Expiry	Timestamp
0024	0071	01	1305	1245
0185	0096	03	1400	1320
….	….	….	….	….

Intra-link Table

AnteriorNode	PosteriorNode	Expiry	Timestamp
0061	0024	1350	1310
0079	0185	1335	1312
….	….	….	….

Path Table

DestZone	DestNode	Route	NumHops	Expiry	Time
A	0071	0034,0024	3	1305	1250
E	0087	A,C,D	4	1440	1425
….	….	….	….	….	….

An inter-link table stores inter-links discovered by intra-zone route discovery. The term *inter-link* in RZRP is defined as a zone-level link that connects two neighbouring zones via a direct node-to-node connection where both ends of the connection are located at different zones. A gate node is a node that has direct connection with the other nodes locating outside of its local zone.

An intra-link table of a node stores the information of node level connections with its 1-hop away neighbours inside the local zone.

A path table contains two types of path: Intra-path and Inter-path. An intra-path is a node-to-node path that starts from the node owning this table to one of its neighbouring zones. An inter-path is a zone-to-zone manner path. It only exists at nodes that belong to one of the intermediate zones on the path from a source zone to a destination zone. An intra-path is constructed from the links in the intra-link table via some graphic search algorithms (e.g. Dijkstra's shortest-path algorithm). An inter-path can be cached directly from inter-zone RREP packet.

C. *Link Expiry Time Prediction*

The notations used in this section are defined in **TABLE I**.

TABLE I: NOTATION USED IN PREDICTION

Notation	Definition
n_i, n_j	The nodes at each end of a link.
l_{ij}	The link connects n_i and n_j
r	The maximum transmission range of node
d_{ij}	The current distance between n_i and n_j
v_i	The speed of n_i
θ_i	The moving direction of n_i
(x_i, y_i)	The location coordinates of n_i
$t_{current}$	The current time
t_{break}	The period from $t_{current}$ until l_{ij} broken

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_i', y_i') and (x_j', y_j') can be calculated by the following formulas:

$$x_i' = x_i - v_i \cdot t_{break} \cdot \cos\theta_i \quad (1) \qquad y_i' = y_i - v_i \cdot t_{break} \cdot \sin\theta_i \quad (2)$$

$$x_j' = x_j - v_j \cdot t_{break} \cdot \cos\theta_j \quad (3) \qquad y_j' = y_j - v_j \cdot t_{break} \cdot \sin\theta_j \quad (4)$$

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

$$d^2 = (x_i' - x_j')^2 + (y_i' - y_j')^2 \quad (5)$$

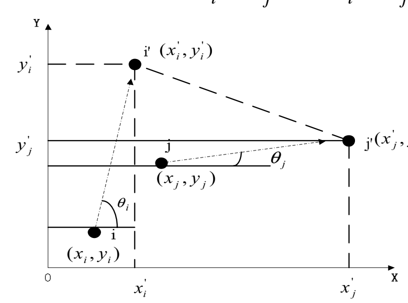

Figure 1: Movements and Positions

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

$$r^2 \leq (x_i' - x_j')^2 + (y_i' - y_j')^2 \quad (6)$$

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

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

This also equals to

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

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

$$r^2 \leq (a - b \cdot t_{break})^2 + (c - d \cdot t_{break})^2 \quad (9)$$

By transforming equation (9) to quadratic format we have

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

Therefore, the value of t can be calculated by using the quadratic formula

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

Hence, $t_{current} + t_{break}$ 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 intra-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*.

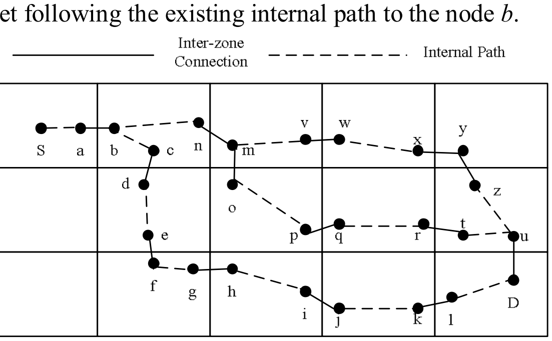

Figure 2: an example of inter-zone route discovery

```
begin
    if received this packet before
        drop this packet
    else if can find the destNode_ID in intra-link table
    or my.Node_ID == destNode_ID
        send inter-zone RREP packet back
    else if can find valid paths to neighbouring zones in path table
        forward this packet
    else if the packet contains my.Zone_ID == false
        initiate intra-zone route discovery
        add my.Zone_ID to the packet
end
```

Procedure 1: Processing inter-zone RREQ

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.

```
begin
    if my.node_ID != packet.destNode_ID
        if received this packet before
            drop this packet
        else if my.inter-link.expiry <packet.expiry
            set packet.expiry=my.inter-link.expiry
            forward packet
    else
        get path from packet
        start data packet transmission
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.

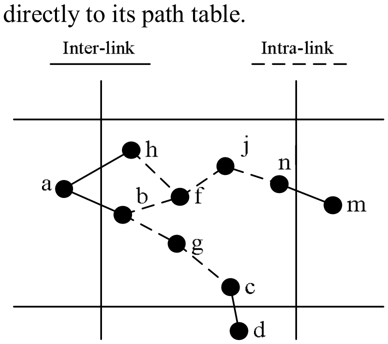

Figure 3: an example of intra-zone route discovery

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.

```
begin
    get the movement information from packet
    predict the link's expiry time
    if received this packet before
```

```
            update my.intra-link_table
            drop this packet
        else if my.Zone_ID != packet.initZone_ID
            generate an intra-zone RREP
            send the RREP back to packet.initiator
            cache the link to my.inter-link_table
        else if my.intra-link_table contains packet.destNode_ID
            overwrite the movement information
            forward this packet to my neighbours
            cache the link and the links in packet.Route_List
             to my.intra-link_table
            send inter-zone RREP back to source node
        else
            overwrite the movement information
            forward this packet to my neighbours
            cache the link and the links in packet.Route_List
             to my intra-link table
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* found 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.

```
begin
    if already received this packet from the previous node
        drop this packet
    else
        cache packet.Links_List.inter-link to my.inter-link_table
        cache packet.Links_List.intra-links to my.intra-link_table
        forward this packet
        generate paths to the initiator of this packet
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_{intra}	An intra-path set to d
S_{inter}	An inter-path set to D
R_{intra}	The selected route to d
R_{inter}	The selected route to D
T_{intra}	The lifetime of R_{intra}
T_{inter}	The lifetime of R_{inter}
P^i_{intra}	the i^{th} path in S_{intra}
P^i_{inter}	The i^{th} path in S_{inter}
N^i_{intra}	The number of hops of P^i_{intra}
N^i_{inter}	The number of hops of P^i_{inter}
l^i	The i^{th} link in path
t^i_l	The lifetime of l^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 inter-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_{intra} -the intra-paths set to d
output: $\{R_{intra}, T_{intra}\}$ -the selected intra-path to d and the lifetime of the path
begin
set $T_{intra} = 0$, $R_{intra} = 0$
for $p^i_{intra} \in S_{intra}$, $i = 1,....,|S_{intra}|$ **do** {
  set $t_p = 0$
  for $l^i \in p^i_{intra}$, $i = 1,......|p^i_{intra}$ **do** {
    if $t_p == 0$ then **set** $t_p = t^i_l$
    else if $t^i_l < t_p$ then **set** $t_p = t^i_l$ }
  if $T_{intra} == 0$ and $R_{intra} == 0$ then
    set $T_{intra} = t_p$, $R_d = p^i_{intra}$
  else if $T_{intra} == t_p$ and $N^i_{intra} < N_{intra}$
    then **set** $R_d = p_i$
  else if $T_{intra} < t_p$ then
    set $T_{intra} = t_p$, $R_{intra} = p^i_{intra}$ }
return T_{intra} , R_{intra}
end

Algorithm 1: Intra-path Selection Algorithm

Inter-path Selection Algorithm
input: S_{inter} -the inter-path set to D
output: R_{inter} -the selected inter-path to D
begin
set $R_{inter} = 0$
for $P^i_{inter} \in S_{inter}$, $i = 1,......,|S_{inter}|$ **do** {
  if $R_{inter} == 0$ then **set** $R_{inter} = P^i_{inter}$
  else if $N^i_{inter} < N_{inter}$ then
    set $R_{inter} = P^i_{inter}$
  else if $N^i_{inter} == N_{inter}$ and $T_{inter} < t_p$
    then **set** $R_{inter} = P^i_{inter}$ }
return R_{inter}
end

Algorithm 2: Inter-path Selection Algorithm

IV. Mathematical Analysis

TABLE III: MATHEMATIC NOTATIONS FOR ANALYSIS

Notation	Definition
n	The total number of nodes
l	The total number of links
z	The total number of zones
r_{intra}	Average length of intra-path
r_{inter}	Average length of inter-path
κ	Transmission request rate per second
c	The probability of creating routes for a transmission
t_{proc}	Packet process delay
t_{prop}	Packet propagation delay
b	Broadcast interval
T	Simulation time in seconds

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 can be calculated by the following formula

$$D_{RZRP} = Q \times \left(\sum_{i=1}^{z}\left(d_i + \sum_{j=1}^{n \div z} d_{ij}\right) + \sum_{m=1}^{r_{inter} \times r_{intra}} d_m\right) \quad (12)$$

Where $Q = k \times T \times c$ is the total number of route creation requests during simulation time T. $d_i = t_{proc}$ is the delay caused by processing of inter-zone route discovery packet at a zone. $d_{ij} = t_{prop} \times \frac{n}{z}$ is the delay caused by propagating intra-zone route discovery packet in a relay zone. $d_m = t_{prop}$ 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

$$D_{ZHLS} = Q \times \left(\sum_{i=1}^{z} d_i + \sum_{m=1}^{r_{inter} \times r_{intra}} d_m\right) \quad (13)$$

Where $Q = k \times T$ is the total number of route creation requests during the simulation time T. $d_i = t_{proc} + t_{prop} \times \frac{n}{z}$ is the delay caused by reactive searching of destination zone ID at each zone. $d_m = t_{prop}$ 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.

$$H_{RZRP} = Q \times \sum_{i=1}^{z}\left(P_i^{inter} + P_i^{intra}\right) \quad (14)$$

Where $Q = k \times T \times c$ is the total number of route creation requests during simulation time T. $P_i^{inter} = r_{intra}$ is the total number of inter-zone route discovery packet propagated in a zone. $P_i^{intra} = \frac{n}{z}$ 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:

$$H_{ZHLS} = Q \times \sum_{i=1}^{z} q_i + \frac{T}{b} \times \sum_{j=1}^{z}\left(P_j^{NLSP} + P_j^{ZLSP}\right) \quad (15)$$

Where $Q = k \times T$ is the total number of route creation requests during simulation time T. $q_i = r_{intra}$ is the total number of destination zone ID searching packets propagated in a zone. $\frac{T}{b}$ is the number of broadcasting times during T. $P_j^{NLSP} = (n \div z)^2$ is the total number of node link state packets propagated in a zone. $P_j^{ZLSP} = n$ is the total number of zone level link 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.

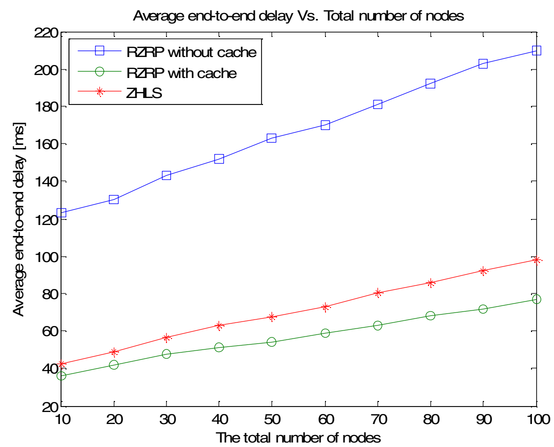

Figure 4: Average end-to-end delay

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 neighbouring 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.

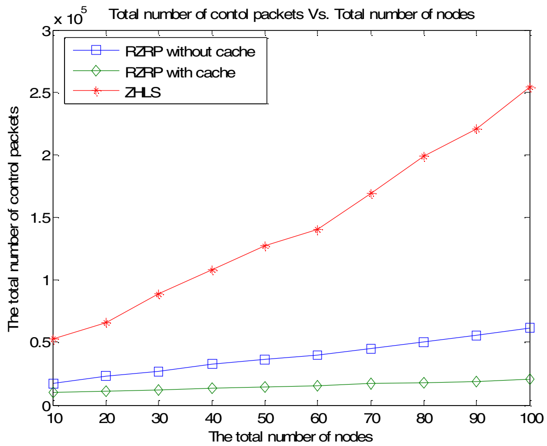

Figure 5: Control packet overhead

Figure 5 shows the impact of the total number of nodes to the total number of control packets that propagated over the whole network. It can be seen that the total number of control packets generated by ZHLS is sharply increased while the number of node increases due to the proactive route information maintenance. On the contrary, both the RZRP implementations are insensitive to the increase of the total number of nodes. However, the RZRP implementation without the cache scheme generates more control packets than the implementation with cache scheme. This is because of that, as the more nodes exist in a zone the more connections will be established to its neighbouring zones, making the connectivity between them more robust. As a result, the number of requests for intra-zone route discovery is reduced.

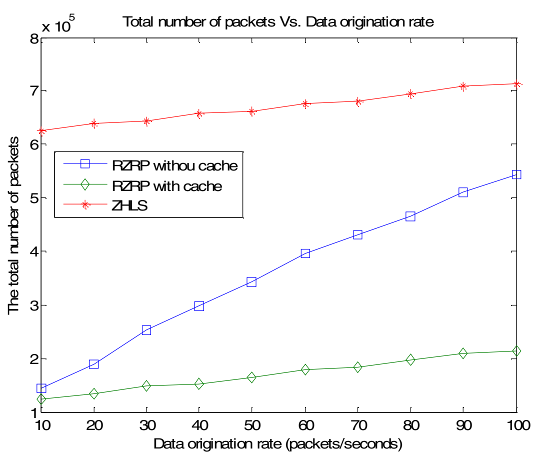

Figure 6: The total number of packets

Figure 6 shows the total number of packets propagating over network as the data origination rate increases. As we can see, the hybrid protocol is less sensitive than reactive protocol to the increase of data origination rate. This is due to the nature of reactive protocol. It must initiates route discovery for each transmission request. However, by using caching scheme in reactive protocol, the paths discovered by previous route discoveries can be reused as long as it remains valid. Such implementation helps reactive protocol remains insensitive to the data origination rate increasing. Moreover, due to the proactive routing information maintenance the hybrid protocol still generates a huge number of control overheads than reactive protocol.

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 deliver 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

[1] M. Joa-Ng and I. -T. Lu, "A Peer-to-Peer Zone-Based Two-Level Link State Routing for Mobile Ad Hoc Networks". *IEEE Journal on Selected Areas in Communication*, 17(8):1415-1425, 1999

[2] W. Liao and J. Sheu and Y. Tseng, "GRID: A Fully Location-Aware Routing Protocol for Mobile Ad Hoc Networks", Telecommunication Systems, 18(1-3): 37-60, 2001

[3] Z. J. Haas and M. R. Pearlman, "The Zone Routing Protocol (ZRP) for Ad-Hoc Networks", *IETF MANET draft,* 2003

[4] P. Samar, M. R. Pearlman, Z. J. Haas, "Independent zone routing: an adaptive hybrid routing framework for ad hoc wireless networks", *IEEE/ACM Transactions*, 12(4): 595 – 608, 2004

[5] T. Park, K-G. Shine, "Optimal Tradeoffs for Location-Based Routing In Large-Scale Ad Hoc Networks", *IEEE/ACM Transactions on Networking, Volume 13 Issue 2,* pp. 398-410, April 2005

[6] C. E. Perkins, Elizabeth M. Belding-Royer, and Ian Chakeres, "Ad Hoc On Demand Distance Vector (AODV) Routing.", *IETF Internet draft*, draft-perkins-manet-aodvbis-00.txt, Oct 2003

[7] D. B. Johnson, D. A. Maltz, and Y.-C. Hu, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks", *IETF Internet Draft*, April 2003

[8] C. E. Perkins, E.M. Royer, S. R. Das, and M. K. Marina, " Performance of Two On-demand Routing Protocols for Ad Hoc Networks", *IEEE Personal Communications, Volume 13 Issue 2,* pp. 3-12, February 2001

[9] R. Beraldi and R. Baldoni, "A Caching Scheme for Routing in Mobile Ad Hoc Networks and Its Application to ZRP", *IEEE Transactions on Computers*, *Vol. 52, No. 8,* pp. 1051-1062, August 2003

[10] W. Lou and Y. Fang, "Predictive Caching Strategy for On-Demand Routing Protocols in Wireless Ad Hoc Networks", *Wireless Networks Volume 8*, pp. 671-679, 2002

[11] W. Su, S-J. Lee and M. Gerla, "Mobility Prediction and Routing in Ad Hoc Wireless Networks", *International Journal of Network Management, Volume 11 Issue 1,* pp. 3-30, January 2001

[12] S. Jiang, D. He and J. Rao, "A Prediction-Based Link Availability Estimation for Routing Metrics in MANETs", *IEEE/ACM Transactions on Networking, Volume 13 Issue 6,* pp. 1302-1312, December 2005

[13] Y. Hu and D.B. Johnson, "Caching Strategies in On-Demand Routing Protocols for Wireless Ad Hoc Networks", *Proc. Mobicom 2000*, pp. 231-242, Aug. 2000

[14] J. Shen, K. Yang, "A Prediction-based Location Update Algorithm in Wireless Mobile Ad-hoc Networks", *Proc. 2005 International Conference on Computer Networks and Mobile Computing (ICCNMC'05), Spring LNCS Series*, pp. 692-701, August 2005