A Location-based Predictive Route Caching Scheme for Pure Reactive Zone-based Routing Protocol in Mobile Ad Hoc Networks

Weibo Chen[1], Kun Yang[1] and Xueqi Cheng[2]
1. University of Essex, School of Computer Science and Electronic Engineering, Colchester, CO4 3SQ United Kingdom
2. Institute of Computer Technology, Chinese Academy of Science, Beijing, P. R. China
Email:{wchen, kunyang}@essex.ac.uk, cxq@ict.ac.cn

Abstract: in mobile ad hoc routing protocols, the control overhead and packet delivery delay are two major metrics for protocol performance. Reactive routing reduces control overhead as it finds routes on demand. On the other hand, it also prolongs data packet delivery delay compared to proactive routing protocols. To leverage this problem, route cache techniques are employed to balance control overhead and data packet delivery delay. In this paper, we propose a reactive location-based predictive caching scheme. The essence of this caching scheme lies in the neat integration of reactive location-based link availability prediction and time-based entry removal mechanism. The integration guarantees that the valid cached routes are never removed while stale routes are removed with the minimum cost of network resources. This caching scheme is embedded in the Reactive Zone-based Routing Protocol (RZRP) and evaluated by a simulation study. After accommodating social information from mobile end users, this mechanism can be well utilized in mobile social networks.

I. Introduction

Mobile social network is a new research area which can be regarded as a group of mobile users conducting social networking on their wirelessly-connected mobile devices in an ad-hoc manner. Routing is a major issue in such networks and it shares similarities with mobile ad hoc networks. Due to node mobility and constrained resources issues, the major challenge in mobile ad hoc routing design is how to maximise the success of data packet delivery at as low cost of network resources as possible under rapid change of network topology. The control overhead and packet delivery delay are two major metrics for protocol performance observation. Proactive routing protocols are developed to accommodate frequent topological changes in the network by periodically exchange route information between nodes. Based on the information, a consistent network topological view that reflects the real up-to-date network situation can be constructed at each node. Hence, routes can be formed before data transmission start. In consequence, the end-to-end packet delivery delay is minimised. However, due to proactive packet exchange, this kind of implementation is not efficient for network resources when the data transmission request at a lower rate. On the other hand, the reactive routing protocols only create routes on-demand by sending out route discovery packets every time when data transmission requests arrive. Such implementation efficiently saves network resources, and improves network scalability. However, the establishment of routes based on network wide packet flooding may lead to "broadcast storm" problem [1]. Moreover, as routes are created on-demand, the packet end-to-end delay takes additional time for establishing route.

Route caching technique that stores newly discovered routes for reusing is one of the popular solutions to improve performance of reactive routing protocols in terms of decreasing route discovery requests as much as possible. By using route cache, both the cost for route discovery and packet end-to-end delivery delay can be reduced. However, the "freshness" of cached entries must be guaranteed. As the cached entries are either removed too early or too late degrades routing performance. Currently, the stale entries can be removed from cache by using packet-based approach or time-based approach. The packet-based approach initiates a notification packet to inform the other nodes in the network. On receiving the notification packet node removes the stale route that the packet mentions. The implementation of packet-based approach can be either proactive or reactive. The proactive implementation of packet-based approach requires a node to report its existence to all neighbours periodically, if a node fails to do so, any route contains this node will be removed from caches of its neighbours. The reactive implementation of packet-based approach requires a node to broadcast out a notification packet only when a route error is detected or it will become unavailable. By using packet-based approach, the stale routes will be removed from cache quickly. However, the broadcast of notification packets may be inefficient to network resources especially in large-scale networks with high mobility situation. In the other hand, the time-based approach associates a Time-To-Live (TTL) value to every cached entry. Route will be removed from cache when the TTL expires. Compare to packet-based approach, the time-based approach is cost efficient and simple to implement. However, the value of TTL must be well tuned to precisely reflect the real existence of routes. As whether the TTL is set too small or too large will result in additional routing latency and traffic overhead. Moreover, the TTL in a caching scheme normally are unique and static, which means that all entries in cache are associated with the same value of TTL and the TTL of each entry may not be able to adjust after it been put into cache.

In this paper, we propose a reactive location-based predictive caching scheme. The major objective of this caching scheme is to improve performance of the Reactive Zone-based Routing Protocol (RZRP) [2] in terms of increasing reliability of routing paths, reducing both the cost of route discovery and packet end to end delivery delay. To achieve this design goal and solve the problems discussed above, this caching scheme integrates location-based link availability prediction and time-based entry removal mechanism together. In this caching scheme, the TTL of every entry in cache is calculated individually by a location-based link availability prediction scheme instead of assigning a static unique TTL to all entries. The TTL of each entry can also be adjusted or recalculated at any time when route error notifications are received. In order to minimise the cost for availability estimation, we piggyback the movement information of node into route discovery packets during a route discovery request cycle. The movement information is limited for one-hop away neighbours only, and will be replaced by a node that rebroadcasts the packet.

The reset of this paper is organised as follows. The preliminary knowledge is presented in section 2. Following section 2, the section 3 describes the operations of this caching scheme in details. After section 3, the caching scheme will be analysed in section 4. Finally, this paper concludes in section 5.

II. Preliminaries

A. Assumptions

As the prediction requires movement information of both ends of a link, we assume that all nodes are equipped with GPS receivers or equivalent devices. Hence, current location coordinates, moving direction, speed, and current time can be obtained directly from such equipments. 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.

B. Overview of the Reactive Zone-based Routing Protocol

The Reactive Zone-based Routing Protocol (RZRP) [2] is a pure reactive zone-based header-less two-level routing protocol proposed by the same authors. It aims to reduce control traffic overhead caused by proactive maintaining local or global neighbourhood routing information and frequent gateway node elections in high mobility and large-scale network situations.

As a zone based routing protocol, a zone in RZRP is defined as a fixed non-overlapped geographic area that pre-partitioned by using location information [3] instead of node connectivity [4]. A unique zone ID is assigned to each zone. Any node that wishes to join the network must be associated with a zone ID by mapping its current location to the zone partition map. The location coordinates of each node will be checked periodically in order to refresh its zone ID. As all zones are pre-partitioned based on their geographical information, these zone IDs can be used to represent current location of nodes, so that it may be transmitted over network instead of transmitting location coordinates. Since the length of zone ID is much smaller than coordinates, the overhead introduced by transmitting location information can be reduced.

The pure reactive implementation of RZRP is guaranteed by implementing two-phase reactive route discoveries: inter-zone route discovery and intra-zone route discovery. The inter-zone route discovery is triggered by source node when a data transmission request arrives and cannot find any valid path to the destination. The inter-zone route discovery has the responsibility to establish routes between source zone and destination zone in zone-to-zone manner. The intra-zone route discovery is triggered when a node first receives an inter-zone route discovery request from its neighbouring zone and cannot find any valid route to destination node or forward the request to neighbouring zones. The responsibility of intra-zone route discovery includes that confirming existence of destination node in a zone, discovering connectivity status inside the zone and establishing paths to neighbouring zones. . Different from that in the inter-zone route discovery, routing paths established by intra-zone route discovery is on a node-to-node basis.

To transmitting data packets, RZRP combines both source routing and next-hop routing together. When transmitting data packets, the source node predefines complete routing path from source zone to destination zone in zone-to-zone manner. On receiving these data packets, intermediate nodes decide which next-hop to go based on their own route information in order to forward the packets towards next relay zone. As authors in [3] proved that zone level connections are more robust than node level connections, this combination improves reliability and flexibility of routing paths in RZRP. According to the performance evaluation in [2], this reactive hierarchical protocol generates less control overhead than hybrid hierarchical protocols. However, as the routing path to destination node and neighbouring zones are created on-demand rather than pre-decided on a periodic basis, RZRP suffers longer end-to-end packet delivery delay than hybrid or proactive hierarchical protocols.

C. Data structures

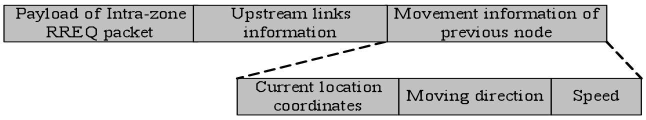
In order to implement this caching scheme, all nodes are required to maintain three tables: connected neighbouring zones table, internal links table, and routing paths table. The structures and purposes of these tables are described as follows.

Connected neighbouring zones table (CNT): a connected neighbouring zones table contains information that reflects connectivity status of local zone with its neighbouring zones. An entry of connected neighbouring zone table is a sequence of <local_gate_nodeID, neighbouring_zoneID, neighbour_gate_nodeID, expiry_time, timestamp>. Where the local_gate_nodeID is a node in the local zone that has direct connection with a node locates at its neighbouring zone. The neighbour_gate_nodeID is the ID of the node that local gate node connects with. The expiry_time field indicates the time when this connection becomes invalid and will be removed from this table.

Routing paths table (RPT): a routing paths table contains two types of path: inter-zone level path and internal node level path. Where the inter-zone level paths are paths in zone-to-zone manner that all discovered by the inter-zone route discovery procedure of RZRP. The internal node level paths are paths in node-to-node manner that either discovered by intra-zone route discovery of RZRP or formed by links from internal links table. An entry of routing paths table is a sequence of <destNode, destZone, nextHop, route, numOfHops, expiry_time, timestamp>.

Internal links table (ILT): The internal links table is a table stores all internal links of a zone that are learned from paths in routing paths table. The purposes of this table are to store local network topological view and construct routing paths to internal nodes that inside the same zone. An entry of internal links table is a sequence of <anterior_NodeID, anterior_ZoneID, posterior_NodeID, posterior_ZoneID, expiry_time, timestamp>. This table contains two types of links: internal link and external link. Internal link is a link that anterior_ZoneID and posterior_ZoneID are the same. External link is a link that both ends are in different zones.

In order to propagate node movement information to next hop neighbours, some additional fields are piggybacked to original control packets of RZRP. Their structures are showed below.



Two additional fields are piggybacked to the original intra-zone RREQ packet. The upstream links information field contains the links and their lifetime from upstream. The movement information field contains the current location coordinates, moving direction, and moving speed of previous node. The movement information filed will be overwrote when current node forward the packet to its next-hop neighbours.

The above format shows two additional fields that are piggybacked to the original intra-zone RREP packet. The lifetime of route field indicates the lifetime of current route to the neighbouring zone, which will be updated during the propagation back to the node that initiated this intra-zone route discovery. The downstream links information field contains all down stream links and their predicted lifetimes.

Payload of Inter-zone RREP packet	Lifetime of the discovered Inter-zone level path

The above format shows an additional field has added to the original inter-zone RREP packet. The lifetime filed indicates the lifetime of the inter-zone level path to destination node, and will be updated by the border nodes of each routing zone it passes through.

D. Link Expiry Time Prediction Method

Table 1: notations in this section

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

The expiry time of a link is the time when both ends of the link travel to the position where the distance between them reaches their maximum transmission range.

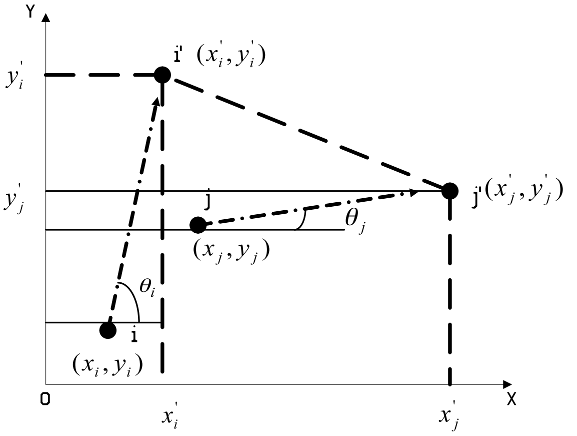


Figure 1 an example of node movement

According to [5], 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:

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

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

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

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

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

This also equals to

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

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

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

By transforming equation (9) to quadratic format we have

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

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

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

Hence, $t_{current} + t_{break}$ is identified as the expiry time of l_{ij}.

III. Operations of the Caching Scheme

The implementation of this caching scheme defines the situations when and where to read, write and update cached route entries.

A. Read Routing Information from Cache

When the source node n_s requires data transmission to the destination node n_d, it implements the following procedure to search its tables for route to n_d.

```
Begin
1. search for the ID of $n_d$ in ILT if ILT contains $n_d$
2.     implement data transmission preparation procedure
3. else search for routes to $n_d$ in RPT
4.     if RPT contains route to $n_d$
5.         implement data transmission preparation procedure
6.     else initiate an inter-zone route discovery request and search CNT
7.         if CNT contains all IDs of neighbouring zones
8.             get routes to neighbouring zones from RPT and forward the
               inter-zone RREQ packet to these zones
9.         else initiate an intra-zone route discovery request
End
```

Procedure 1 cache searching procedure at source node

When an intermediate node n_a in another zone receives the inter-zone RREQ packet from n_s, it implements the Procedure 2 to make the routing decision whether forwarding the packet, sending back a reply message, or initiating an intra-zone route discovery.

```
Begin
if the inter-zone RREQ packet contains the zone ID of this node
    get intra-zone level paths from RPT and put them into $S_{intra}$
    implement intra-zone route selection algorithm
    forward this packet to the selected relay node
else
    search for the ID of $n_d$ in ILT if ILT contains $n_d$
        send inter-zone RREP packet back to $n_s$
    else if CNT contains all IDs of neighbouring zones
        get intra-zone level paths from RPT and put them into $S_{intra}$
        implement next relay node selection procedure
        forward this packet to the selected node
    else initiate an intra-zone route discovery request
End
```

Procedure 2 Cache searching procedure at intermediate node during inter-zone route discovery

Where S_{intra} is a set contains all valid intra-zone paths. Procedure 2 repeats until the inter-zone RREQ packet finally reaches the destination node or an intermediate node that can find valid path to the destination node.

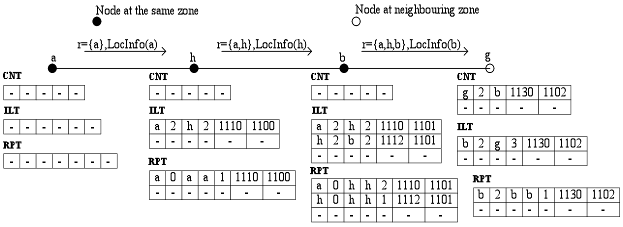
Following procedure is carried out at source node once the routes to destination are established.

```
Begin
search for the ID of $n_d$ in RPT and put the routes to $n_d$ into $S_{inter}$
implement inter-zone path selection algorithm
put the selected inter-zone level path into data packet
search for the ID of local gate node that connects the next relay zone in CNT
search for routes to local gate node in RPT and put the routes into $S_{intra}$
implement intra-zone path selection algorithm
forward the packet to next-hop relay node of the selected route
End
```

Procedure 3 Packet transmission preparation at source node

Where S_{inter} is a set contains all valid inter-zone paths.

B. Write Route Information into Cache



a: CNT

-	-	-	-	-

a: ILT

-	-	-	-	-	-

a: RPT

-	-	-	-	-	-	-

h: CNT

-	-	-	-	-

h: ILT

a	2	h	2	1110	1100
-	-	-	-	-	-

h: RPT

a	0	a	a	1	1110	1100
-	-	-	-	-	-	-

b: CNT

-	-	-	-	-

b: ILT

a	2	h	2	1110	1101
h	2	b	2	1112	1101
-	-	-	-	-	-

b: RPT

a	0	h	h	2	1110	1101
h	0	h	h	1	1112	1101
-	-	-	-	-	-	-

g: CNT

g	2	b	1130	1102
-	-	-	-	-

g: ILT

b	2	g	3	1130	1102
-	-	-	-	-	-

g: RPT

b	2	b	b	1	1130	1102
-	-	-	-	-	-	-

Figure 2 an example of writing information into tables during intra-zone RREQ packet propagation

Figure 2 shows an example that how routing information is written into the routing tables of intermediate nodes during the propagation of intra-zone RREQ packet. When an intermediate node receives an intra-zone RREQ packet, it implements following procedure.

```
Begin
get the movement information of previous node from the movement information of previous node field
get links and their lifetimes from the upstream links information field and cache them into ILT
implement the prediction method to predict the lifetime of the link to previous node
get the initiator zone ID from the packet
if the initiator zone ID == my zone ID
    add the link to previous node with the predicted lifetime into the upstream links information field and cache them into ILT
    replace the movement information field with own movement information
    send the intra-zone RREQ packet to all next hop neighbours
    build paths to upstream nodes bases on the cached upstream links in ILT
    add paths to upstream nodes into RPT
    forward this packet to next hop neighbours
else
    send an intra-zone RREP packet back to initiator
    cache the link to previous node to CNT
    build path to previous node and put it into RPT
End
```

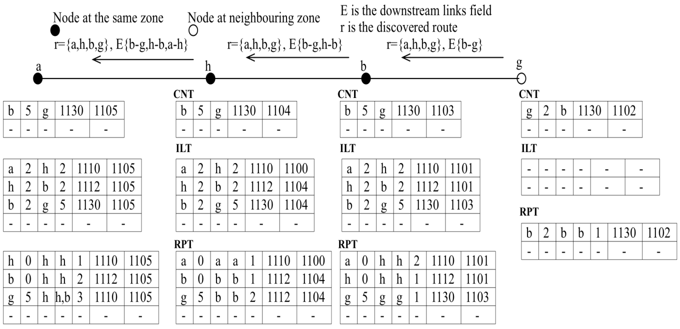
Procedure 4 Write information into cache at intermediate node on receiving intra-zone RREQ

When a border node receives an intra-zone RREQ packet from its neighbouring zone, it will send back an intra-zone RREP packet back. On receiving an intra-zone RREP packet, the intermediate nodes implement following procedure in order to write downstream routing information into routing tables.

```
Begin
get the zone ID and node ID of the replier from packet
get all downstream links from the downstream links information field
search for the link that connects the replier
put the link and the zone ID of replier into CNT
put the rest downstream links into ILT
if the packet intend for this node
    drop the packet
else
    add the link to previous node and its predicted lifetime into downstream links information field
    forward the packet to next hop node
End
```

Procedure 5 Write information into cache at intermediate node on receiving intra-zone RREP

After the intra-zone RREP reaches the initiator, all the intermediate nodes should cache all upstream and down stream routing information into their routing tables just as Figure 3 below.



a:

b	5	g	1130	1105
-	-	-	-	-

a	2	h	2	1110	1105
h	2	b	2	1112	1105
b	2	g	5	1130	1105
-	-	-	-	-	-

h	0	h	h	1	1110	1105
b	0	h	h	2	1112	1105
g	5	h	h,b	3	1110	1105
-	-	-	-	-	-	-

h: CNT

b	5	g	1130	1104
-	-	-	-	-

h: ILT

a	2	h	2	1110	1100
h	2	b	2	1112	1104
b	2	g	5	1130	1104
-	-	-	-	-	-

h: RPT

a	0	a	a	1	1110	1100
b	0	b	b	1	1112	1104
g	5	b	b	2	1112	1104
-	-	-	-	-	-	-

b: CNT

b	5	g	1130	1103
-	-	-	-	-

b: ILT

a	2	h	2	1110	1101
h	2	b	2	1112	1101
b	2	g	5	1130	1103
-	-	-	-	-	-

b: RPT

a	0	h	h	2	1110	1101
h	0	h	h	1	1112	1101
g	5	g	g	1	1130	1103
-	-	-	-	-	-	-

g: CNT

g	2	b	1130	1102
-	-	-	-	-

g: ILT

-	-	-	-	-	-
-	-	-	-	-	-

g: RPT

b	2	b	b	1	1130	1102
-	-	-	-	-	-	-

Figure 3 An example of writing information into tables during intra-zone RREP propagation

C. Cache Maintenance

The maintenance of routing tables includes update, remove, and repair entries. As described early, the removal of stale routing information from tables mainly based on the individually calculated Time-To-Live (TTL) of each entry. However, the prediction is not hundred percent accurate due to different node mobility models and accuracy of movement information that GPS receiver provides. In this case, the scheme must have the ability to dynamically adjust the TTL of each individual entry in tables.

The routing tables can be updated by route error packets or during another route discovery request cycle. Following algorithm shows the procedure that a node receives a route error packet.

```
Begin
get movement information from the error packet
implement link availability prediction.
if the prediction result doesn't equals to the TTL of the entry in ILT
```

```
4.    update the new predicted result to the link in ILT
5.    recalculate the TTLs of paths that contains the updated link in RPT
6. replace the movement information and send the packet to next node
End
```

Procedure 6 Cache update procedure on receiving route error packet

As Procedure 6 shows, the route error packet contains the newest movement information of both ends of the broken link. Hence, on receiving the error packet all downstream and upstream intermediate nodes along the routing path could update their routing tables.

IV. Performance Evaluation

Table 2 Notations used in this section

k	Transmission request rate per second
c	The possibility of creating route for a data transmission request
n_{total}	The total number of nodes
z_{total}	The total number of zones
l_{total}	The total number of links
$r_{\text{int}\,ra}$	Average length of intra-zone routing path
$r_{\text{int}\,er}$	Average length of inter-zone routing path
t_{proc}	Packet process delay
t_{prop}	Packet propagation delay
T	Simulation time in seconds

D. Communication overheads

By implementing this caching scheme, RZRP could efficiently reduce communication overheads in terms of saving the total number of route discovery requests and route repair requests. The efficiency can be represented by the formula below

$$C_{overall} = Q \times \sum_{i=1}^{r_{\text{int}\,er}} (\varphi_i^{\text{int}\,ra} + \varphi_i^{\text{int}\,er}) \qquad (12)$$

where $Q = k \times T \times c$ is the factor that represents the total number of route discovery requests generated during the simulation. $\varphi_i^{\text{int}\,ra} = n_{total} \div z_{total}$ is the total number of intra-zone route discovery packets propagate in a zone. $\varphi_i^{\text{int}\,er} = r_{\text{int}\,ra}$ is the total number of times that the inter-zone route discovery packet is relayed in a zone.

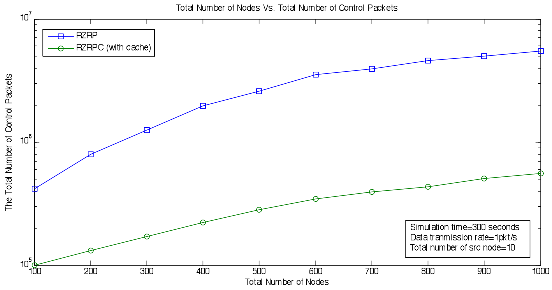


Figure 4 Control overhead Vs. Total number of nodes

Figure 4 shows the communication overhead that generated by two different implementations of RZRP while the total number of nodes increasing. As it can be seen from the figure, the trends of both curves are nearly the same, which both increased while the total number of nodes increasing. The only difference between them is that, the implementation of RZRP with this caching scheme generates less overhead as it doesn't need to initiate route discovery for every data transmission request.

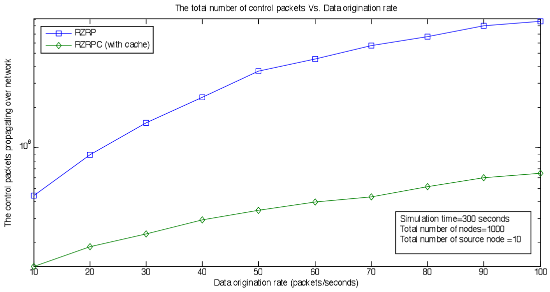


Figure 5 Control overhead Vs. Data originate rate

As Figure 5 indicates, the overhead generated by RZRP without implementing this cache scheme is sharply increased while the data originate rate increasing. It is due to the reactive nature that RZRP has to initiate route discovery request for every data transmission request.

E. Packet End-to-End Delivery Delay

The packet end-to-end delivery delay includes both of the time for route creation and the time of packet propagation. By implementing this cache scheme, the time for route creation can be reduced. Hence, the packet end-to-end delay is reduced, which can be represented by the following formula

$$D = Q \times (\sum_{i=1}^{z} (d_i + \sum_{j=1}^{n \div z} d_{ij}) + \sum_{m=1}^{r_{\text{int}\,er} \times r_{\text{int}\,ra}} d_m) \qquad (13)$$

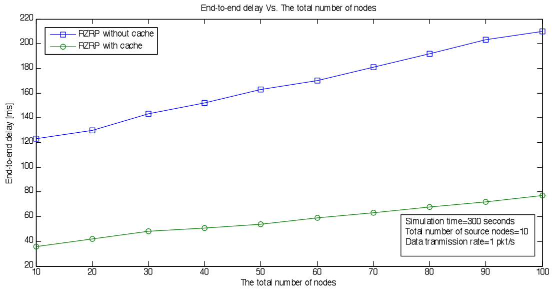


Figure 6 Packet end-to-end delay Vs. Total number of node

As shown in Figure 6, the implementation of RZRP without cache suffers longer delay. This is understandable in that reactive protocols usually have longer end-to-end delay as the routing path to destination node and neighbouring zones are created in on-demand manner. However, by implementing this predictive caching scheme, the reduction of control overhead is visible.

V. Conclusion

This paper has proposed a reactive location-based cache scheme that works underneath the RZRP in terms of improving its routing performances. The novelty of this cache scheme lies in the integration of location-based link availability prediction and time-based removal mechanism. In this caching scheme, all stale entries will be removed from routing tables based on their existing time TTL that individually calculated and dynamically adjusted during route discovery cycles by the prediction method. In order to minimize the network resource consumption for the prediction, the movement information of node is piggybacked to intra-zone route discovery packet. Therefore, this cache scheme guarantees that valid cached routes are never removed while stale routes are removed with the minimum cost of network resources. Our future work will introduce social information and knowledge into the above proposed algorithms to guide routing discovery/selection and caching in the context of mobile social networks.

Acknowledgments

The work of this paper was funded in part by the K. C. Wong Education Foundation, Hong Kong.

References

[1] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The Broadcast Strom Problem in a Mobile Ad Hoc Network", Prco. Mobicom '99, pp. 151-162, Aug. 1999.

[2] W. Chen, K. Yang. "RZRP: A Pure Reactive Zone-based Routing Protocol with Location-based Predictive Caching Scheme for Wireless Mobile Ad Hoc Networks", Proc. of IEEE Int. Conf. on Wireless Communications and Mobile Computing (IWCMC 2008), 6-8 August 2008, Crete, Greece. Page(s):505 - 510

[3] 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.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[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.