ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, No.11, November 2009, pp.3086−3100
doi: 10.3724/SP.J.1001.2009.00583

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

分簇覆盖的移动自组织网中节点位置辅助路由算法*

王 毅[1+], 董 梁[2], 梁涛涛[1], 杨新宇[1], 张德运[1]

[1](西安交通大学 电子信息与工程学院,陕西 西安 710049)

[2](Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119260, Singapore)

Cluster-Based Location Aided Routing Algorithm for Mobile Ad Hoc Networks

WANG Yi[1+], DONG Liang[2], LIANG Tao-Tao[1], YANG Xin-Yu[1], ZHANG De-Yun[1]

[1](School of Electrical and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

[2](Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119260, Singapore)

+ Corresponding author: E-mail: Joshua_wy@126.com

Wang Y, Dong L, Liang TT, Yang XY, Zhang DY. Cluster-Based location aided routing algorithm for mobile ad hoc networks. *Journal of Software*, 2009,20(11):3086−3100. http://www.jos.org.cn/1000-9825/583.htm

Abstract: Using location information to assist routing is often proposed as an efficient means to achieve scalability in large mobile ad hoc networks (MANET). This paper proposes an algorithm, named as Cluster-Based Location Aided Routing (CLAR), a scalable and efficient routing algorithm for MANET. CLAR runs on top of a one-hop cluster cover of the MANET, which can be created and maintained by, for instance, the Least Cluster Change (LCC) algorithm. It has been proven that LCC can maintain a cluster cover with a constant density of clusterheads with the minimal update cost. CLAR then utilizes nodes' location information to improve the network layer performance of routing. The location information of destination node is used to predict a smaller isosceles triangle, rectangle, or circle request zone, which is selected according to the relative location of the source and the destination, that covers the estimated region where the destination may locate. Instead of searching the route in the entire network blindly, CLAR confines the route searching space into a much smaller estimated range. Simulation results have shown that CLAR outperforms other protocols significantly in route set up time, routing overhead, mean delay and packet collision, and simultaneously maintains low average end-to-end delay, high success delivery ratio, low control overhead, as well as low route discovery frequency.
Key words: mobile ad hoc network; location aided; routing algorithm; cluster

摘 要: 在大规模移动自组织网中,利用节点位置信息辅助建立路由被认为是一种有效提高无线网络路由可扩展性的方法.提出了一种可扩展与高效的、适用于移动自组织网络的路由算法——分簇覆盖的节点位置信息辅助路由算法(CLAR).CLAR使用如最小簇改变(LCC)算法,建立并保持的单跳分簇结构为拓扑.已有文献证明,LCC是更新成本最小的保持簇头节点密度均一的分簇算法.CLAR 利用网络节点的位置信息提高无线网络路由的网络层性能,由

* Supported by the National High-Tech Research and Development Plan of China under Grant No.2006AA01Z210 (国家高技术研究发展计划(863)); the Singapore Academic Research Fund (AcRF) (新加坡教育科研基金); the NUS Academic Research Fund (新加坡国立大学教育科研基金)

Received 2008-08-26; Accepted 2009-05-05

目标节点的位置信息预测并构造一个较小的、形状为等腰三角形、矩形或圆形的区域.该区域需保证覆盖目标节点可能存在的位置,且根据源节点与目标节点间的相对位置决定该区域适宜的形状,从而限制源节点在一个较小的“请求域”内寻找可用路由,而不是在网络内盲目寻找.仿真实验结果表明,与其他路由算法相比较,CLAR 路由算法在路由建立所需时长、路由代价、平均时延及数据包冲突等参数上表现优良.同时,算法保持了低平均时延、高数据包到达率、低控制开销及低路由寻找次数等优势.

关键词: 移动自组织网;位置信息辅助;路由算法;分簇结构

中图法分类号: TP393 **文献标识码:** A

1 Introduction

Developing routing protocols for MANET has been an extensive research area during the past few years, and significant progress has been made in developing the algorithms and algorithm refinements to achieve scalable routing[1−11]. Many existing routing protocols (FSR[6], LANDMAR[7], DSR[8], CBRP[9]) are designed to scale in networks of a few hundred nodes. These protocols use the known connectivity relation with mobile nodes' neighbors to do route discovery blindly, which always produces dense routing traffic and collision. It does not only waste a large portion of wireless bandwidth, but also costs more route construction time. In order to reduce the routing overhead, the location-based routing protocol, which uses the location information of mobile nodes to assist the routing task has attracted more attention. This idea is to use location information in order to reduce propagation of control messages (LAR[3]), to control packet flooding (DREAM[10]), to reduce intermediate system functions or to make simplified packet forwarding decisions (GSR[11]). However, the heavy communication overheads for location updates make such an approach hard to support large-scale ad hoc networks. On the other hand, the node location information is obtained either from intermediate nodes or from the desired nodes on request basis. Clearly, this approach has low cost for exchanging location information but high cost for searching. However, the stability is usually weak, since there is neither a location map maintained in each node nor specific location servers available.

This paper focuses on a hierarchical cluster-based location-aided routing protocol for the mobile nodes in MANET with good scalability and efficient route discovery. Different from some typical structures being used in fixed geographic mobile networks, the clustering structure is self-organized and adaptable to build a location-aided routing protocol. In brief, our approach is to form a one-hop clustering (cluster cover) of the network and then to perform route discovery by forwarding route requests over the necessary participated nodes. More specifically, we employ the Least Cluster Change (LCC)[12] algorithm to establish and maintain a clustering structure of the network, whereby a node in a given cluster can reach the head of the cluster in one hop. When a source node wants to send a message to a destination node, the source node uses the location information of destination node to predict an adaptable request zone. More precisely, the location information of the source, the destination and the expected zone is utilized to predict an isosceles triangle, rectangle or circle request zone that reduces the coverage of route discovery space and covers the position of the destination. The smaller route discovery space reduces the total number of route discovery messages exchanged and the probability of collision. If the destination node is reached, it responds with an acknowledgment and the route discovery is completed. Otherwise, when route discovery failed or the route hole existed, an increasing exclusive search approach is used to redo route discovery by a progressive increasing request zone.

2 Clustering as a Basis for Routing

We consider a wireless system consisting of homogeneous nodes which are distributed on a flat two-dimension field. Each node has a GPS receiver and the geographical position can be measured. Nodes are uniquely identified, i.e., by using the MAC addresses (ID). Any specific mobility model has been assumed, although simulations are conducted for the Random Way Point (RWP) mobility model.

The CLAR builds upon some specific properties of the underlying clustering structure. We employ the Least Cluster Change (LCC) algorithm for one-hop clustering. The choice of the LCC clustering algorithm is motivated by the pervious work[12], where the authors have proved that the LCC algorithm is asymptotically optimal or near-optimal with respect to: (1) the number of clusters maintained and (2) the cost of an update. More advantages of using LCC for clustering have been investigated in Ref.[12]. In LCC, a CH may not communicate with other adjacent CHs directly but it needs some nodes, called gateways (GWs), to relay message. These nodes, CHs and GWs, establish a virtual backbone, which plays a key role in routing as it simplifies the routing process to one in a smaller sub-network. Using the virtual backbone nodes, messages are mainly exchanged between them, instead of being broadcasted to all the nodes, and the route discovery process is speeded up. Thus, the virtual backbone is able to reduce the routing overhead, to minimize the routing delay and to simplify the connectivity management[13].

3 Cluster Overlay Location Aided Routing Algorithm

In this section, we describe the Cluster-based Location-Aided Routing (CLAR) algorithm which runs on top of a cluster cover of the MANET.

3.1 Propagation of information

Initially, in mobile ad hoc network environments, a node may not know the GPS location (either current or old) of other nodes. However, similar to the LAR[3], as time progresses, each node can get location information for many nodes either as a result of its own route discovery or as a result of message forwarding for another node's route discovery. For instance, if *S* includes its current location in the route request message, and if *D* includes its current location in the route reply message, then each node receiving these messages can know the current locations of not only the nodes *S* and *D*, but also the relay nodes. Besides, once a node receives these messages, the location information of the nodes participated in the routing process can be updated. In general, the location information may be propagated by piggy-backing it on any packet. Similarly, a node also propagates to other nodes the information about its mobility (or some other measure of speed). In our simulations, we set the default maximum speed of node for $v_{\max}$=30m/s, and that is known to all nodes.

3.2 Expected zone

In the route discovery procedure, the source *S* uses the location information of the destination *D* to estimate the region where *D* expects to appear, and the region is called expected zone[3]. We extend the definition of expected zone in LAR, because many literatures have proved that the expected zone cannot be calculated exactly[1,2] with the method proposed in LAR. However, when the route request message arrives at the original location of *D*, after some time, this time interval can be called Δt. As shown in Fig.1, in order to calculate a more exact expected zone, we must take the time interval Δt into account. Assume that *S* knows that *D* is at location $GPS(x_D,y_D)$ at time t_0, and the current time is t_1. There are two scenarios: (1) the routing from *S* to *D* has been established. *S* knows the transmission time of a message from *D* to *S*, so the Δt can be estimated as the transmission time from *D* to *S*; (2) the routing from *S* to *D* is not established. In this case, *S* knows the *D*'s location $GPS(x_D,y_D)$, its location $GPS(x_S,y_S)$, and node's maximum speed $v_{\max}$. Then $\Delta t=\sqrt{(x_S-x_D)^2+(y_S-y_D)^2}\,/\,v_{\max}$. Finally, the radius of expected

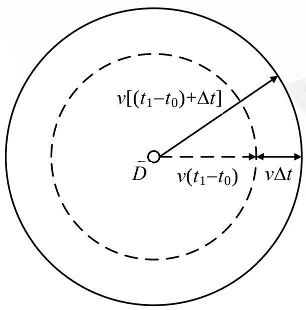


Fig.1 Expected zone

zone is $v_{max}[(t_1-t_0)+\Delta t]$.

Particularly, if after some time, node *S* does not know the previous or the current location of node *D*, then *S* cannot reasonably determine the expected zone—in this case, the entire region that may potentially be occupied by the network is assumed to be the expected zone. And our algorithm is reduced to the basic flooding algorithm. Further, having more information regarding mobility of a destination can result in a smaller expected zone.

3.3 Request zone

Instead of searching the route in the entire network blindly, CLAR confines the route searching space into a smaller estimated region, which is defined as request zone. A node forwards a route request only if it belongs to the request zone. To enhance the probability that the route request will reach the destination, the request zone should not only include the expected zone but also other region around the routing path. It is mainly due to the fact that there is no guarantee that a path can be found only consisting of the nodes in a chosen request zone. Therefore, if a route is not discovered within a suitable timeout period, our protocol allows *S* to initiate a new route discovery with an expanded request zone, which is similar to the hole problem in Section 3.6. In this case, however, the latency in determining the route to *D* will be longer (as more than one round of route discovery will be needed).

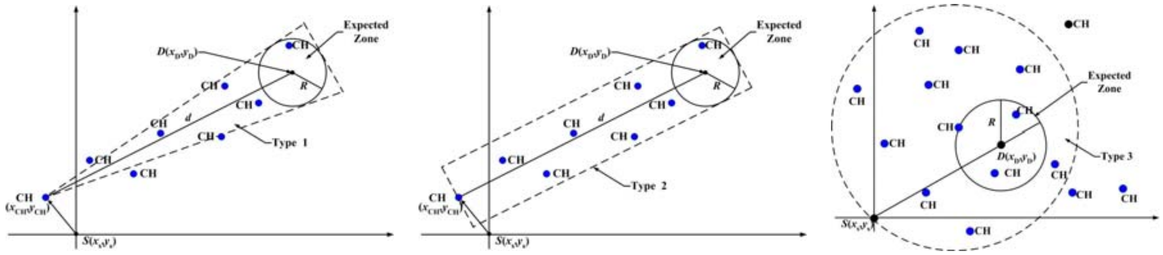
3.4 Selection of request zone

Generally, the accuracy of request zone (i.e., probability of finding an available route to the destination) can be improved by increasing the size of request zone (i.e., total number of nodes contained in this zone). Because the more nodes participates in the routing process, the more probable the establishing route path from *S* to *D* is, and the more reliable the route path is. However, with the size of the request zone increasing, some performance metrics such as total times of packet collision, route set up time and route discovery overhead maybe get worse, and meanwhile, another metric, probability of route recovery, maybe get less. Thus, there exists a trade-off between the performance metrics and the accuracy of request zone (and the size of request zone). In Ref.[3], authors table a proposal that many forms of request zone, such as the circular-shaped, the rectangular- shaped, and the cone-shaped, can be used. As an extension of LAR, to improve the routing performance, CLAR algorithm should select some different types of request zones corresponding to the relation of relative location among the source *S*, the destination *D*, and the expected zone *EZ*. In CLAR, the definition of request zone can be classified as: (scenario I) *S* is outside of expected zone, and *S* and *D* are in different clusters; (scenario II) *S* is outside of expected zone, and *S* and *D* are in the same cluster; (scenario III) *S* is within the expected zone. The types of request zones we introduced are listed as: the isosceles triangle, the rectangle, and the circle. To select the appropriate type of request zone according to the relation of relative location among *S*, *D* and *EZ*, we conduct a series of simulations, in which the configuration is the same as those in Section 4. To examine the performance, we introduce four metrics: (1) the total times of collision, which we define as the total times of collision taking place when using different types of request zone; (2) the route set up time, which we define as the average time required to construct a path to *D*; (3) the route discovery overhead, which we define as the total number of packets transmitted per node per route established from *S* to *D*; (4) the probability of route recovery, which we define as the times of route recovery due to the link failure in each round and the denominator is the total times of route discovery.

3.4.1 Scenario I

Figure 2 shows the cases when the request zone is defined as the isosceles triangle, the rectangle, or the circle respectively. In LCC, the CHs are connected to form a virtual backbone. The connected virtual backbone plays a key role in exchanging the messages between the CHs, instead of being flooded to all the nodes. Thus, it provides an efficient approach to minimizing the flooding traffic during route discovery and speeding up this process as well.

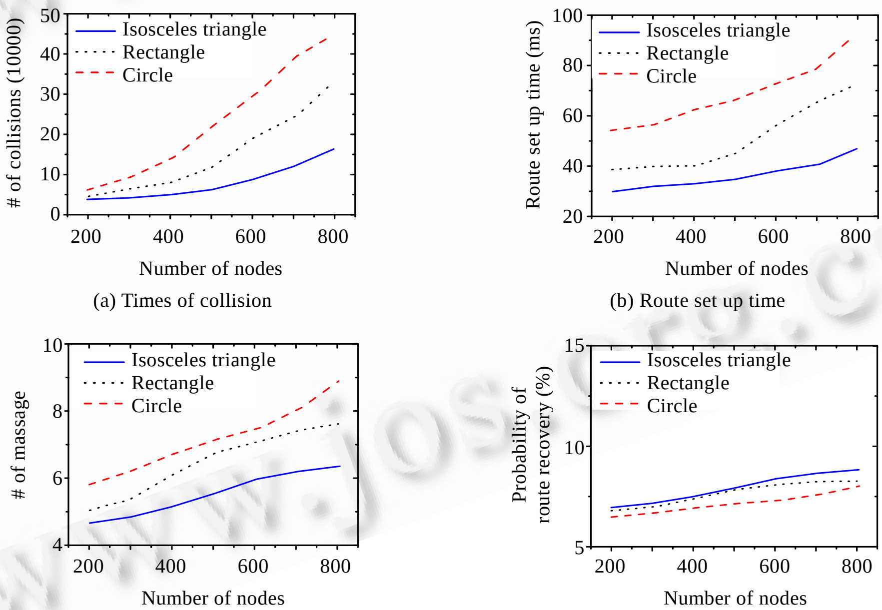
With the network layer performance in consideration, the impact of this characteristic cannot be ignorable.



(a) Isosceles triangle (b) Rectangle (c) Circle

Fig.2 Three types of request zones in scenario I

From Fig.3, it is shown that the metrics, in the case when the type of request zone defined as the isosceles triangle, perform much better than those in the other two. Generally, the probability of collision and the route discovery overhead are proportional to the number of nodes involved in the route process. As shown above, the area of the isosceles triangle is the smallest of the three, so that the smallest number of nodes participates in the route process. Meanwhile, it restrains the messages to forward along the narrowest space. It means that the request message is forced to propagate in as straight a direction as possible. This is preferable in providing a higher chance to select a shorter route. Further, because the relative distance between S and D is quite long, the Δt as well as the area of EZ is not small, the area of the isosceles triangle is large enough to include enough nodes to mitigate the probability of link failure, so that it is profitable to decrease the probability of route recovery.



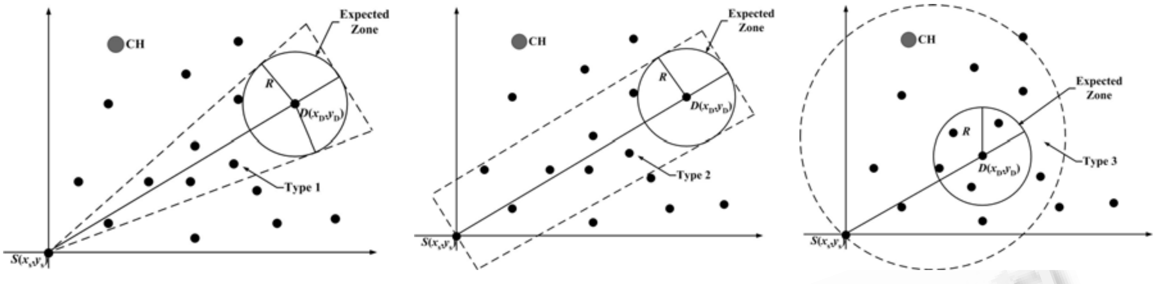
(a) Times of collision (b) Route set up time

(c) Route discovery overhead (d) Probability of recovery

Fig.3 Network layer performance comparison in RZI

3.4.2 Scenario II

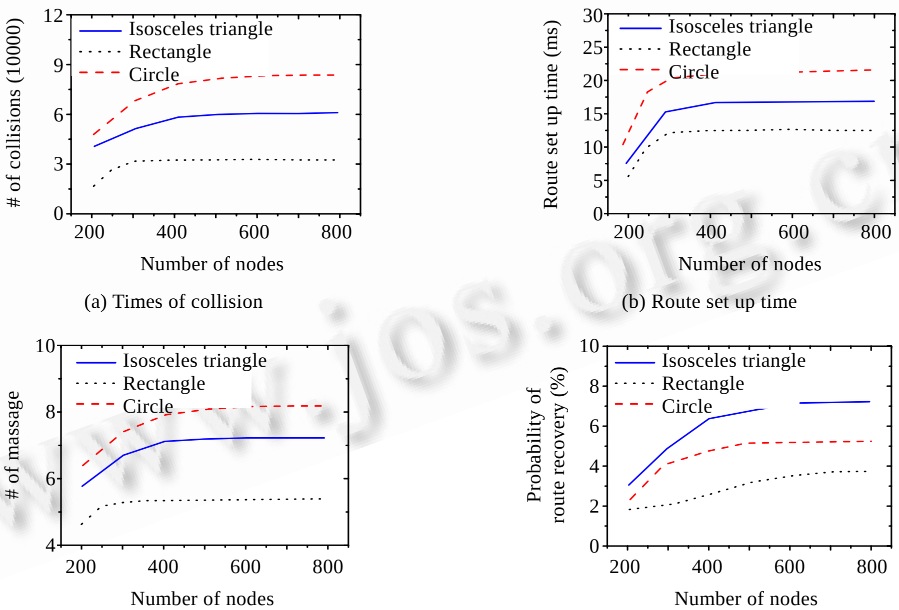
Figure 4 shows the cases when the request zone is defined as the isosceles triangle, the rectangle, or the circle respectively, in Scenario II. As mentioned in Ref.[12], LCC maintains a cluster cover with a constant density of CHs, that means each CH can reasonably support only a certain number of nodes to ensure efficient MAC functioning. Thus, with the node density increasing, the number of CHs is also increasing. However, the number of clustermembers (CMs) in any cluster increases over a certain threshold and then keeps steady. As a result, each performance metric will level off after the node density increases over a certain threshold.



(a) Isosceles triangle (b) Rectangle (c) Circle

Fig.4 Three types of request zones in scenario II

In Fig.5, the results show that in the case when the type of request zone defined as the rectangle, the four metrics are much better than those of the other two. The area of the rectangle (i.e. the number of nodes in the area) is larger than that of the isosceles triangle, but smaller than that of the circle. Because of the mobile characteristic of nodes, too small area (i.e. few nodes) may incur great probability of link failure. If a route is broken or cannot be found, *S* will conduct the route recovery procedure or initiate a new route discovery, it is quite obvious that these actions cause more routing traffic and occupy more network resources. Compared with the circle-shaped situation, there are fewer nodes involved in the routing process in the rectangle-shaped situation. The fewer nodes participates in the process, the fewer packets is transmitted simultaneously, and the less probable of collision increase is and the smaller the discovery overhead is.



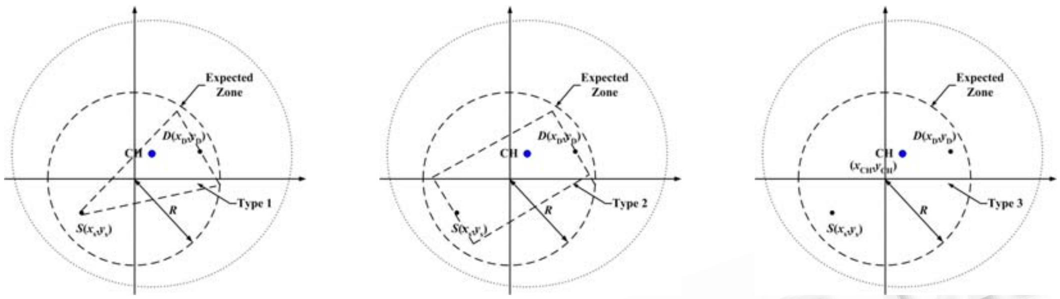
(a) Times of collision (b) Route set up time

(c) Route discovery overhead (d) Probability of recovery

Fig.5 Network layer performance comparison in RZII

3.4.3 Scenario III

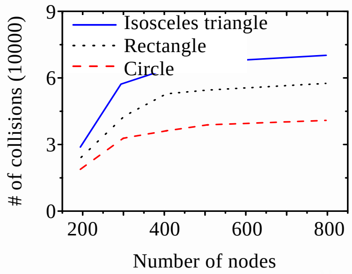
Figure 6 shows the cases when the request zone is defined as the isosceles triangle, the rectangle, or the circle respectively, in Scenario III. S and D are within the expected zone, the relative distance between S and D maybe is less than $v_{max}[(t_1-t_0)+\Delta t]$. This characteristic indicates that the relative distance between S and D is quite small, which affects the four performance metrics greatly. Furthermore, as mentioned in Section 3.4.2, the limitation of the number of CMs in a cluster still works.



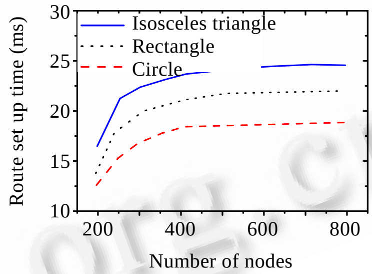
(a) Isosceles triangle (b) Rectangle (c) Circle

Fig.6 Three types of request zones in scenario III

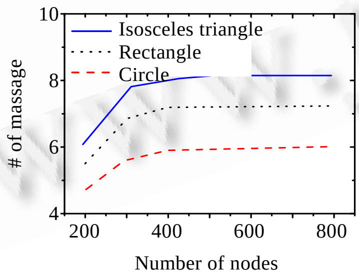
We observe from Fig.7 that the four performance metrics, in the case when the type of request zone defined as the circle, are better than those in the other two. It seems like that the simulation results are in contrast with the analysis above, because the area of the circle is the largest of the three, and the performance in that case should be the worst. According to the simulation data, the main influence on the algorithm performance is due to the link failure. As mentioned above, the relative distance between S and D is quite small. If in the two cases when the type of request zone defined as the isosceles triangle and the rectangle, it is possible that too few relay nodes or no relay node exist because of the narrow route discovery space and node movement, which is easier to make a link failure. This behavior is detrimental to the network performance.



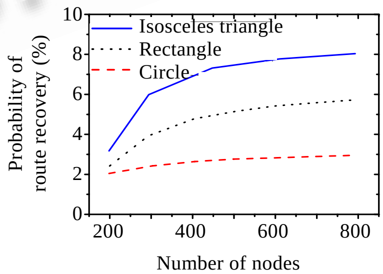
(a) Times of collision



(b) Route set up time



(c) Route discovery overhead



(d) Probability of recovery

Fig.7 Network layer performance comparison in RZIII

3.5 Scenarios of request zone

According to the analysis above, we define the shape of request zone as the isosceles triangle, the rectangle, the circle corresponding to the Scenario I, Scenario II, Scenario III, respectively.

3.5.1 Scenario I

If S is outside of the expected zone, and S and D are in different clusters, then the request zone is defined as a isosceles triangle, named RZI. As mentioned above, the virtual backbone maintained by LCC has been constructed, which always plays a key role in routing as it simplifies the routing process. Intuitively, the virtual backbone nodes can provide "short cut". CLAR should utilize them for remote destination nodes to reduce the transmission delay. The critical factor in Scenario I is that the restricted region should provide more chances to make the request message route go through the connected CHs, more precisely, the virtual backbone nodes, as many as possible.

Different from the request zone RZII or RZIII started by the node S, RZI is started by the CH which S joins. The request message is forwarded from S to its CH; CH checks it and starts the RZI procedure. Thus, RZI includes the current location of CH and the estimated expected zone EZ.

In Fig.8, the RZI corners are CH (whose location is $GPS(X,Y)$), A and B. The area of RZI can be calculated as follows:

$$S_{\Delta}=(d+R)^2\tan\alpha=\frac{\{v_D[(t_1-t_0)+\Delta t]\}(\sqrt{(X-x_D)^2+(Y-y_D)^2}+v_D[(t_1-t_0)+\Delta t])^2}{(X-x_D)^2+(Y-y_D)^2-\{v_D[(t_1-t_0)+\Delta t]\}^2} \tag{1}$$

In CLAR, if D is a CM, we can route the request message to its CH directly, because the CH manages D and communicates with the connected virtual backbone nodes easier. With the backbone nodes, the routing message is exchanged between the CHs, instead of being broadcasted to all the nodes. Thus, in Scenario I, the algorithm always makes effort to make the routing path pass as many CHs as it can.

(a) S is closer to D (b) CH is closer to D

Fig.8 Request zone I

3.5.2 Scenario II

If S is outside of the expected zone, and S and D are in the same cluster, then the request zone is defined as a rectangle, named RZII, including the current location of S and the estimated expected zone EZ. As shown in Fig.9, the area of RZII, whose corners are S, A, B, C and E can be calculated as follows:

$$S_{ABCE}=2R(d+R)=2R(\sqrt{(x_S-x_D)^2+(y_S-y_D)^2}+R) \tag{2}$$

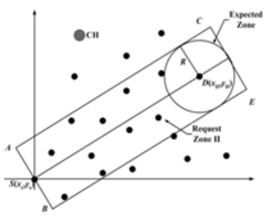
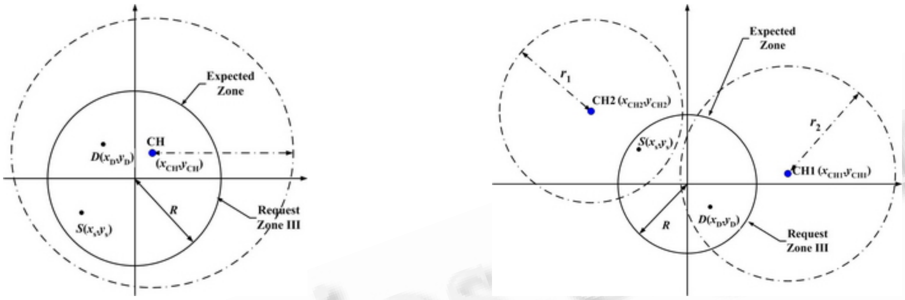


Fig.9 Request zone II

3.5.3 Scenario III

If S is within the expected zone, the request zone is defined as a circle, named RZIII, which is equal to the expected zone. S locally sends the request message in the RZIII. In particular, if and only if the Scenario III occurs, whatever S and D are in the same cluster or adjacent clusters, as shown in Fig.10, S and D can ignore the hierarchical layers temporarily. It means that S can send request message directly without being passed through CHs. Because this method avoids the nodes exchanging the messages with the CHs, it is quite obvious that it can improve the performance.



(a) S, D in same cluster (b) S, D in adjacent clusters

Fig.10 Request zone III

3.6 Hole problem

Because of the narrow space of each request zone, if there are holes in the request zone, the route discoveries are influenced and likely to be repeated many times, which in turn increases the routing overhead and extends the delay of routing path. To overcome the problem, a route hole detection method is proposed.

In Scenario I, S forwards the request message to any relay node i, i checks if the next hop neighbor nodes are located within in the RZI by using the information recorded in its NT. If there is no neighbor node suitable for the next hop, i returns the "Msg_Node_RErr" to S, which includes the location information of neighbor j which i considers as suitable for the next hop. Having received the message, S will increase the angle α to α' and recalculate RZI. As shown in Fig.11(a), the line through $CH(x_{CH},y_{CH})$ and $j(x_j,y_j)$ is given by

$$Y=\frac{y_{CH}-y_j}{x_{CH}-x_j}(X-x_j)+y_j \tag{3}$$

Similarly, line through CH and D is given by

$$Y=\frac{y_{CH}-y_D}{x_{CH}-x_D}(X-x_D)+y_D \tag{4}$$

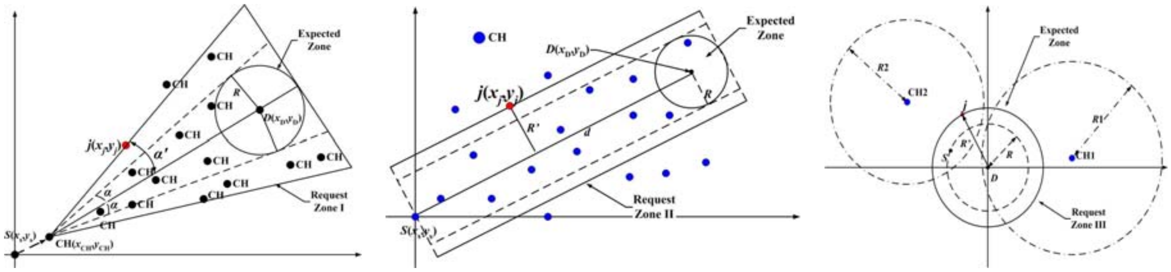
Thus, the new α' can be calculated as follows:

$$\alpha'=\arctan\left|\frac{\left(\frac{y_{CH}-y_j}{x_{CH}-x_j}\right)-\left(\frac{y_{CH}-y_D}{x_{CH}-x_D}\right)}{1+\left(\frac{y_{CH}-y_j}{x_{CH}-x_j}\right)\left(\frac{y_{CH}-y_D}{x_{CH}-x_D}\right)}\right| \tag{5}$$

In Scenario II and Scenario III, the methods we proposed are similar. The idea is to enlarge the coverage of request zone. For instance, in Scenario II, after receiving the message Msg_CM_RErr, S will extend the line segment SA to SA', as shown in Fig.11(b). The new R' can be calculated as follows:

$$R'=\frac{|(x_D-x_S)x_j+(y_S-y_D)y_j+(x_Sy_D-y_Sx_D)|}{\sqrt{(x_D-x_S)^2+(y_D-y_S)^2}} \tag{6}$$

In Scenario III, S will extend the radius. The new radius r' is the distance from j to D, as shown in Fig.11(c).



(a) New RZI (b) New RZII (c) New RZIII

Fig.11 New request zone

3.7 Route discovery

When the source S wants to transmit a data packet to the destination D, it firstly estimates the expected zone and request zone, and then performs a series of operations to establish the routing path.

Step 1. S calculates an expected zone by the approach we described in Section 3.2, while it uses the basic information of D which is exacted from the information $\langle CH\_ID(D),ID(D),GPS\rangle$. Meanwhile, the estimated relative distance between S and D, $d(S,D)$, can be obtained.

Step 2. S judges the scenario:

```
procedure Judge_scenario
begin
  if (d(S,D)≤v_max[(t_1−t_0)+Δt])
      then scenario III is initiated;
          return RZIII;
  else if (CH_ID(S)==CH_ID(D))
      then scenario II is initiated;
          return RZII;
      else scenario I is initiated;
          return RZI;
end
```

Step 3. S defines a request zone to include the expected zone according to the result above.

Step 4. S sends a route request message, named "Msg_Route_RReq", that includes the information of the RZ and D, whose options are $\langle type,sour,dest,RZ(X),pathlist,h,routed\rangle$. S sets $type$=$request$, $sour$=$ID(S)$, $dest$=$ID(D)$, $pathlist$=$null$,h=0, $routed$=0, and the value of X in $RZ(X)$ can be decided based on procedure Judge_scenario in Step 2.

Step 5. After received the "Msg_Route_RReq", an intermediate node i invokes this process. The pseudo-code of each procedure is given below:

```
procedure Establish_path
begin
   Receive(Msg_Node_RReq);
   if (i∉RZ)
      then Discard(Msg_Node_RReq);
   exit(0);
```

```
    else if (i∈RZ)
      then case X of
        1: call Routing_RZI; break;
        2: call Routing_RZII; break;
        3: call Routing_RZIII; break;
end
procedure Routing_RZI
begin
    if ([Trial mode]==CH)
      if (D is i's CM)
      then add i to pathlist; h++; routed=1;
      else
      then select next hop m from i's NT;
            and CH is preferred;
      call Judge(j);
    if (i==CM)
      if (D is in i's NT)
      then add i to pathlist; h++; routed=1;
      else
      then select next hop n from i's NT;
            and CH is preferred;
      call Judge(j);
end
procedure Routing_RZII
begin
    if (CH_ID(i)==CH_ID(D))
      if (D is in i's NT)
      then add i to pathlist; h++; routed=1;
      else
      then select next hop m from i's NT;
      call Judge(j);
    else
      then i drops the route request;
end
procedure Routing_RZIII
begin
    then select next hop j from i's NT;
    call Judge(j);
end
procedure Judge(j);
begin
    if (j∉RZ)
```

```
        then return Msg_Node_RErr to S;
    else if (j∈RZ)
        then send Msg_Node_RReq to j;
            add node i to pathlist; h++;
end
```

Step 6. When “Msg_Route_RReq” has been received by D, it unicasts a route reply message along the reverse direction of the route recorded in the request packet to S. If D has received multiple pieces of route request message, D chooses the one with the least h to reply. The route reply message, named “Msg_Route_RRly”, whose options are $\langle CH\_ID(D),ID(D),GPS(x_D,y_D),pathlist\rangle$.

Step 7. S waits for receiving the rout reply message from D. Then, the routing path from S to D is established.

3.8 Route recovery

If a route failure is detected by an intermediate node in the routing path, or the source S does not receive any reply message within a suitable time period, the route must be recovered as soon as possible. If the route failure is detected by an intermediate node, there are two methods to repair the route. The first method is to initiate a route discovery process by the broken node, called local search, to repair the broken path. This method is investigated in Ref.[1], and it can reduce the overhead of route recovery as well as the latency of route rediscovery. If the local search method fails, the second method should be employed. The second method is that the node detected the route failure sends back a route error message “Msg_Route_Fail” to inform the source that a route failure has occurred. After receiving the message, the source re-initiates a route discovery to search for a new routing path.

4 Performance Evaluation

In this section, we present simulation results to illustrate the performance of CLAR protocol. The simulator was implemented within Global Mobile Simulation (GloMoSim) 2.03 library by C++ language[14]. The GloMoSim library is a scalable simulation environment for mobile wireless network using parallel discrete-event simulation capability provided by PARSEC[14]. We conduct a series of simulations to examine the comparison of the network layer performance of the well-known CBRP, VSR, LAR1, and TZRP with CLAR.

In our simulation, all network nodes are located in a physical area of size 1000m by 1000m to simulate actual mobile ad hoc networks. The size of network is in the range of [200,300,400,500,600,700,800] nodes that were generated according to a uniform distribution. The mobility model selected is the Random Waypoint model (RWP). We conduct simulations for the RWP mobility models with a randomly distributed speed in the range from 5~30m/s; the pause time is fixed to 30 seconds. The propagation path loss model is the TWO-RAY model that uses free space path loss (2.0,0.0) for near sight and plane earth path loss (4.0,0.0) for far sight. The radio bandwidth of each mobile node is 2Mbps. Following Ref.[1], we assumed that different frequency bands for the intra-cluster communication inside the individual clusters and the inter-cluster communication among adjacent cluster leaders. Our simulation model considers the distributed coordination function (DCF) of 802.11, which employs carrier sense multiple access with collision avoidance (CSMA/CA). The simulation time of each round lasts for 1000 seconds. Each simulation result is obtained from an average of the all simulation statistics.

CLAR is compared with some famous and classical protocols, such as CBRP[9], VSR[4], LAR1[3] and TZRP[5]. Four performance metrics are introduced to evaluate the routing performance of CLAR:

(1) Average end-to-end delay: The end-to-end delay is averaged over all surviving data packets from the source S to the destination D;

(2) Success delivery ratio: Ratio of the total number of data packets delivered to the destination D to those

generated by the source *S*;

(3) Route discovery frequency: The total number of route discoveries initiated per second;

(4) Control overhead: The total number of routing control packets normalized by the total number of received data packets.

Figure 12 shows the results of average end-to-end delay. From Fig.12, CBRP shows a fast increase in packet end-to- end delay. The reason is that when there is a large amount of control packets contenting for channel usage, the data packets have to back off a lot for a free slot. VSR usually has large routing packets but fewer control packets than CBRP, so the delay is shorter than VSR. The packet end-to-end delay in LAR1 increases slightly because the location of a node is constantly updated via location_update messages sent by the moving node and therefore changes in the topology have little effect on the delay. TZRP uses two zones to limit the nodes involved in the route discovery, and reduces the control packets. CLAR performs much better than other four protocols in more "stressful" (i.e. larger number of nodes, more load), that is greatly contributed to the establishment of the request zone and three routing strategies of request zone we proposed.

Figure 13 shows the results of success delivery ratio for the five. It illustrates that CLAR outperforms at any mobility speed, especially exhibited higher performance at higher speed. From Fig.13, when the mobility speed=30m/s, because CLAR has two methods to recover the failure path, it always loses fewer packets than CBRP, VSR, LAR1 and TZRP: 41.52%, 37.48%, 19.78%, 24.51%, respectively. The results demonstrate that CLAR may provide efficient fault tolerance in the sense of faster and efficient recovery from route failures in dynamic networks.

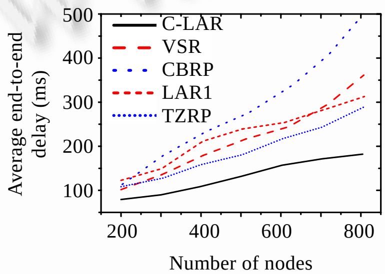


Fig.12 Average end-to-end delay vs. number of nodes

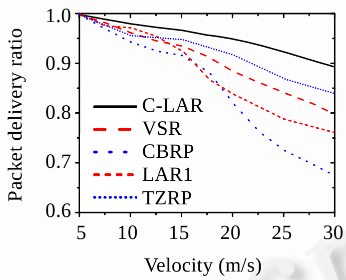


Fig.13 Success delivery ratio vs. max.velocity

Figure 14 shows the results of discovery frequency performance. CLAR needs less discovery times to maintain these routing paths. CBRP is a simple path routing protocol based on cluster, so the source must broadcast a lot of discovery packets to recover the broken path. VSR cannot use the local search mechanism to repair the broken path, but always waits for the source node's response. Thus, VSR also has more discovery times than that of CLAR. Both LAR1 and TZRP use locality information to reduce the route discovery frequency. LAR1 relies on a location update mechanism that maintains approximate location information for all nodes in a distributed fashion. While nodes moving, the approximate location information is constantly updated. TZRP uses Crisp zone for proactive routing and efficient broadcasting, and a Fuzzy Zone for heuristic routing using imprecise locality information. The results demonstrate that a desirable property of CLAR that the routes still remain with high probability even at high rates of mobility. It is interesting to observe that the effects of the parameters in the clustering algorithm on this metric.

Figure 15 shows the results of control overhead as a function of the node mobility in RWP mobility model. The control overhead includes that route request packet and route reply packet for a node involved in the routing process. Further, to CLAR, the overhead for propagating information and establishing cluster structure is still included. The total number of overhead per node among five protocols increases when the nodes move fast. With a higher mobility of nodes, the topology of network changes faster, so the control overhead also increases. The simulation results show that the control overhead of CLAR is lower than that of CBRP, VSR, LAR1 and TZRP, especially when the node number increases fast enough. By comparison, we can notice from Fig.15 that the larger the size of the network is, the lower the control overhead of CLAR is relative to the other four protocols.

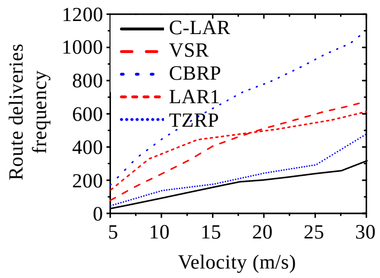


Fig.14 Route discovery frequency vs. max.velocity

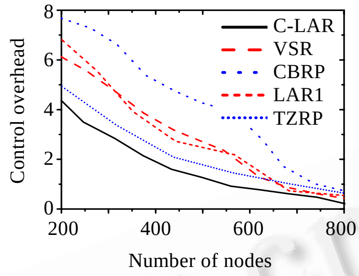


Fig.15 Control overhead vs. number of node

5 Conclusions

Using the location information of mobile nodes to assist routing can confine the route searching space into a smaller estimated range. The smaller space to be searched, the less routing overhead and broadcast storm problems will occur. In this paper, we have developed the Cluster-based Location-Aided Routing (CLAR) algorithm for mobile ad hoc networks. CLAR discovers routes with the location information of the source node and the destination node on the cluster-based network of mobile ad hoc networks. The location information of the source, the destination and the expected zone is utilized to predict an isosceles triangle, rectangle or circle request zone that reduces the coverage of route discovery space and covers the position of the destination. This approach limits the search for a route to the so-called request zone, which is determined by the expected location of the destination node at the time of route discovery. Furthermore, an increasing-exclusive search approach is proposed to redo the route discovery if the previous route discovery fails. It guarantees that the areas of route rediscovery will never exceed twice the entire network. The simulations show that CLAR outperforms other routing algorithms in many metrics, e.g., route set up time, routing overhead, mean delay and packet collision, and maintains low average end-to-end delay, high success delivery ratio, low control overhead and low route discovery frequency. In the aspect of energy consumption, the above metrics are all very important for power-saving. Therefore, CLAR can save more power and lengthen system lifetime.

Meanwhile, CLAR runs on top a cluster cover of the network with a constant density of clusterheads, which Ref.[12] have proven to be maintained by the Least Cluster Change (LCC) algorithm. This clustering architecture has the following advantages in support of mobile ad hoc networks: First, it is able to significantly reduce the overheads with the use of simple cluster level route combined with self-determined inter-cluster forwarding based on cluster mobility pattern or group characteristics. Next, the clustering architecture has flexible scalability in support of large scale ad hoc networks. Third, it has the capability of cluster-level self-route recovery against interlink or route failures. Finally, based on the distance effect, it is able to provide more accurate location information within the cluster and nearby neighborhoods, which matches the dynamic nature of ad hoc networks very well.

References:

[1] Ritchie L, Yang HS, Richa AW, Reisslein M. Cluster overlay broadcast (COB): MANET routing with complexity polynomial in source-destination distance. IEEE Trans. on Mobile Computing, 2006,5(6):653−667.

[2] Eriksson J, Faloutsos M, Krishnamurthy SV. DART: Dynamic address routing for scalable ad hoc and mesh networks. IEEE ACM Trans. on Network, 2007,15(1):119−132.

[3] Ko YB, Vaidya NH. Location-Aided routing (LAR) in mobile ad hoc networks. Wireless Networks, 2000,6(4):66−75.

[4] Theoleyre F, Valois F. Virtual structure routing in ad hoc networks. In: Lee YK, Lee BG, Lee KT, Yim CH, eds. Proc. of the IEEE Int'l Conf. on Communications 2005. Seoul: IEEE Press, 2005. 3078−3082.

[5] Wang L, Olariu S. A two-zone hybrid routing protocol for mobile ad hoc networks. IEEE Trans. on Parallel and Distributed Systems, 2006,15(12):1105−1116.

[6] Iwata A, Chiang CC, Pei GY, Gerla M, Chen TW. Scalable routing strategies for ad-hoc wireless networks. IEEE Journal of Selected Area in Communications, 1999,17(8):1369−1379.

[7] Pei GY, Gerla M, Hong XY. LANMAR: Landmark routing for large scale wireless ad hoc networks with group mobility. In: Perkins CE, Toh CK, Vaidya NH, eds. Proc. of the IEEE/ACM 1st Annual Workshop on Mobile and Ad Hoc Networking and Computing 2000. Boston: IEEE Press, 2000. 11−18.

[8] Johnson DB, Maltz DA, Broch J. DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks. Ad Hoc Networking. Addison-Wesley, 2001. 139−172.

[9] Jiang ML, Li JY, Tay YC. Cluster based routing protocol (CBRP) functional specification. IETF Internet-Draft: draft-ietf-manet-cbrp-spec-01.txt, 1998.

[10] Basagni S, Chlamtac I, Syrotiuk VR, Woodward BA. A distance routing effect algorithm for mobility (DREAM). In: Osborne WP, Moghe D, eds. Proc. of the 4th Annual ACM/IEEE Int'l Conf. on Mobile Computing and Networking (MobiCOM'98). Dallas: IEEE Press, 1998. 76−84.

[11] Jain R, Puri A, Sengupta R. Geographical routing using partial information for wireless ad hoc networks. IEEE Personal Communication, 2001,8(1):48−57.

[12] Chiang CC, Gerla M. Routing and multicast in multi-hop, mobile wireless network. In: Acampora A, ed. Proc. of the 4th IEEE Int'l Conf. on Universal Personal Communications 1997. San Diego: IEEE Press, 1997. 197−221.

[13] Huang H, Richa AW, Segal M. Approximation algorithms for the mobile piercing set problem with applications to clustering in ad hoc networks. Mobile Networks and Applications, 2004,9(2):151−161.

[14] Bagrodia R, Meyer R, Takai M, Chen Y, Zeng X, Martin J, Park B, Song H. Parsec: A parallel simulation environment for complex systems. IEEE Computer, 1998,31(10):77−85.

WANG Yi was born in 1981. He is a Ph.D. at the Xi'an Jiaotong University. His current research areas are wireless networks and mobile computing.

YANG Xin-Yu was born in 1973. He is a Postdoctoral and professor at the Xi'an Jiaotong University and a CCF senior member. His current research areas are wireless networks and web services.

DONG Liang was born in 1973. He is a senior scientist at the National University of Singapore. His search areas are wireless networks and audio & video processing.

ZHANG De-Yun was born in 1941. He is a Professor at the Xi'an Jiaotong University. His current research areas are wireless networks and mobile computing**.**

LIANG Tao-Tao was born in 1985. He is a Master at the Xi'an Jiaotong University and a CCF senior member. His current research areas are wireless networks and web services.