PAPER

Cluster Based Location-Aided Routing Protocol for Large Scale Mobile Ad Hoc Networks

Yi WANG[†a)], ***Student Member***, **Liang DONG**[††*b)], ***Member***, **Taotao LIANG**[†], **Xinyu YANG**[†], ***and*** **Deyun ZHANG**[†], ***Nonmembers***

SUMMARY Routing algorithms with low overhead, stable link and independence of the total number of nodes in the network are essential for the design and operation of the large-scale wireless mobile ad hoc networks (MANET). In this paper, we develop and analyze the Cluster Based Location-Aided Routing Protocol for MANET (C-LAR), a scalable and effective routing algorithm for MANET. C-LAR runs on top of an adaptive cluster cover of the MANET, which can be created and maintained using, for instance, the weight-based distributed algorithm. This algorithm takes into consideration the node degree, mobility, relative distance, battery power and link stability of mobile nodes. The hierarchical structure stabilizes the end-to-end communication paths and improves the networks' scalability such that the routing overhead does not become tremendous in large scale MANET. The clusterheads form a connected virtual backbone in the network, determine the network's topology and stability, and provide an efficient approach to minimizing the flooding traffic during route discovery and speeding up this process as well. Furthermore, it is fascinating and important to investigate how to control the total number of nodes participating in a routing establishment process so as to improve the network layer performance of MANET. C-LAR is to use geographical location information provided by Global Position System to assist routing. The location information of destination node is used to predict a smaller rectangle, isosceles triangle, or circle request zone, which is selected according to the relative location of the source and the destination, that covers the estimated region in which the destination may be located. Thus, instead of searching the route in the entire network blindly, C-LAR confines the route searching space into a much smaller estimated range. Simulation results have shown that C-LAR 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 networks, routing, location-aided, clustering*

1. Introduction

Scalable routing is one of the key challenges in designing and operating large scale mobile ad hoc networks [1]–[4]. In order to ensure effective operation as the total number of nodes in the MANET becomes large, the overhead of the employed routing algorithms should be low and independent of the total number of nodes in MANET. 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 MANET routing. Among them, DSR [5], AODV [6], TORA [7], ZRP [8], TZRP [9], LAR [10], GPSR [11], SLURP [12] are well-known classic algorithms. Recently, further approaches for routing on top of a cluster cover of a set of core or bone nodes have been proposed in [13]–[19]. Yet, some key challenges remain in the development of scalable MANET routing algorithms. In particular, the existing MANET routing algorithms have been formally analyzed either:

- for the route discovery, a total elapsed time or total number of messages exchanged that depend on the overall network size, such as the total number of nodes in the MANET or the total diameter of the network (see, for instance, [5]–[7]), or
- with restrictive assumptions about the overall network topology, such as limiting the network density (see, for instance, [8]–[12]).

For these reasons, the existing MANET routings are of limited use for large scale MANET consisting of a large number of nodes and having a large diameter.

Typically, when wireless network size (the total number of mobile nodes) increases beyond certain thresholds, common "flat" routing schemes become infeasible because of link and processing overhead [4]. One way to solve this problem and to produce scalable and efficient solutions is hierarchical routing, which is based on the idea of organizing nodes in groups and then assigning nodes different functionalities inside and outside of a group. Both routing table size per node and the total number of update packets transmitted by the nodes are reduced because only part of the network is included, thus control overhead is reduced [13], [15]. However, the hierarchical routing always outperforms locally, when the routing path crosses long range, the number of nodes participating are still large. The cost of an independent node is reduced by using hierarchical routing mechanisms, but the total overhead of MANET may still be great. The efficient method is that the number of nodes involved in the routing process should be controlled by some aided techniques. One possibility direction is to use location information provided by Global Position System (GPS). Instead of searching the route in the entire network blindly, location-aided routing protocol uses the location information of mobile nodes to confine the route searching space

Manuscript received June 23, 2008.
Manuscript revised November 27, 2008.

[†]The authors are with School of Elec. and Info. Engineering, Xi'an Jiaotong University, P.R. China.
[††]The author is with Institute for Infocomm Research, 21 Heng Mui Keng Terrace, 119613 Singapore.
[*]Presently, with Healthcare department, Philips Research Asia-Shanghai, China.
a) E-mail: joshuawy@gmail.com
b) E-mail: ldong@i2r.a-star.edu.sg

DOI: 10.1587/transinf.E92.D.1103

into a smaller estimated range. The smaller route searching space to be searched, the less routing overhead and broadcast storm problem will occur.

The main contribution of this work is to propose a scalable and effective routing protocol for MANET - the Cluster Based Location-Aided Routing Protocol for MANET (C-LAR). C-LAR intends to utilize nodes' location information to improve 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 the destination node may locate. Instead of searching the route in the entire network blindly, 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. C-LAR 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, determined based on 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 when the previous route discovery failed. It guarantees that the areas of route rediscovery will never exceed twice the entire network. The comparison of our algorithm and LAR is studied through extensive simulation. The simulations show that C-LAR 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.

Meanwhile, C-LAR runs on top a cluster cover of the network with some specific properties. In particular, we present a cluster algorithm for one-hop clustering of MANET. In the clustering, we propose a weight-based distributed clustering algorithm which takes into consideration the number of nodes a clusterhead (CH) can handle reasonably (considering node capacity and throughput), mobility, relative distance, battery power and link stability of the nodes. This algorithm assigns node-weights based on the suitability of nodes acting as CHs and the election of the CH is done on the basis of the largest weight among its neighbors. After formatting the clusters, some clustermembers (CMs) are selected to be as the gateway nodes (GWs) to connect the CHs. Then, we use the distributed backbone formation algorithm, which has been presented in [28], to construct a virtual backbone architecture.

2. System Model

We consider a wireless system consisting of homogeneous nodes which are distributed on a flat two-dimension field and let N denote the number of nodes. Each node has a GPS receiver and the geographical position can be measured. We assume that nodes are uniquely identified, i.e., using the MAC addresses (ID). All nodes have the same maximum communication range R and can communicate only if their relative distance is below R. The nodes are with the capability to transmit with some different transmission ranges, which can, for instance, be achieved by power control [23]. We assume that the link between two adjacent nodes, which can communicate with each other directly, is bi-directional. The route path is selected among the one hop neighbors with "symmetric". Therefore, in C-LAR, selecting the route automatically avoids the problems associated with data packet transfer on uni-directional links such as the problem of not getting an acknowledgment for the data packets at each hop.

We consider the problem of unicast routing in the MANET. In particular, we focus on the problems of (1) clustering the entire network, (2) discovering route from a source node S to a destination node D and (3) delivering a message M from S to D. In our analysis, we do not assume any specific distribution of the nodes. However, our route discovery algorithm–as any other algorithm–can only find a route if the network is connected, i.e., if there exists at least one feasible route from the source to the destination node. We do not assume any specific mobility model in our analysis, although we conduct simulations for the Random WayPoint mobility model. We only initially assume, as is reasonable and common, that the mobility of the nodes is on a time scale slower than the route discovery [15].

3. Clustering as a Basis for Routing

The C-LAR protocol is implemented on the top of the clustering structure. Since the entire network is partitioned into smaller logically separated clusters, a CH is elected for each cluster to maintain cluster membership information. The main advantage is that it is easy for a CH to keep track of mobile nodes in its radio coverage, nodes joining or leaving, cluster's topology, and so on. Inter-cluster routes are discovered dynamically using the cluster membership information kept at each CH. By clustering nodes into groups, the protocol efficiently minimizes the flooding traffic during route discovery and speeds up this process as well. Furthermore, the protocol takes into consideration the existence of bi-directional links and uses these links for both intra-cluster and inter-cluster routing. In this section, we present the node clustering in detail.

3.1 System Parameters

Because of the mobile characteristic of the nodes in MANET, to obtain better hierarchical structure, five issues

should be investigated first: (1) the upper bound of the number of CHs, which will yield high throughout but incur as low delay as possible; (2) the relative speed, by which we can get good knowledge about the relative speed between any two nodes; (3) the relative distance, which indicates the communication cost; (4) the residual energy, if the nodes have various battery power to start with, then it would be a more accurate metric to eliminate some nodes from the CH candidate set; (5) the link stability, it means how stable a wireless link between any node pair is. Based on the analysis above, in order to elect the suitable nodes to be as the CHs, five parameters should be introduced: (1) the upper bound of the number of CHs n^*; (2) the average relative speed ξ; (3) the average relative distance θ; (4) the average residual energy ε; (5) the average link stability μ.

3.1.1 Upper Bound of the Number of CHs n^*

To obtain the system parameter (1), the capacity and throughput are considered as the key factors. Generally, capacity is defined as the maximum possible information transfer rate over a channel. The metric, Carrier-to-Interference ratio (C/I), determines directly the capacity of the radio channel and the ad hoc network, consequently. Throughput measures the number of bits per second delivered over the medium, and it is affected by the routing and the offered traffic at each node. If a route cannot be found from a source to a destination, the throughput between these two nodes is virtually zero. Additionally, the offered traffic at one node determines the expected amount of relay traffic and the throughput at other nodes. In the analysis below, a route between the source and the destination always can be found if it exists.

The amount of interference in a MANET is directly related to the output traffic produced per node. The output traffic per node consists of the mobile node's own traffic (we will call this traffic the self traffic) and the traffic that the node relays for other nodes (the relay traffic). Because of relay traffic, the total amount of traffic per node is strongly related to the multi-hop characteristics of the MANET. Our basic assumption here is that the self traffic generated by the mobile nodes is Poisson distributed and independent of each other. All nodes are similar and have the same traffic generation behavior. In other words, mean generated self traffic per node per time interval is the same. We denote the mean value of self traffic per time-slot per node by λ. The length of each time-slot is denoted by t_s. The average number of packet arrivals per unit time is then λ/t_s. Because we assumed a Poisson arrival process, for the probability of k packets arrival during a time interval of length t we have:

$$P_k(t,\lambda) = \frac{(\lambda t/t_s)^k}{k!} e(-\lambda t/t_s) \tag{1}$$

Consider two nodes i and j. When the expected average hop count is $E[h]$, there are in average $E[h]-1$ relay nodes between any source and any destination. i may be a relay node for j with the probability $(E[h]-1)/(N-1)$, and the expected value for relay traffic arriving at i from j is then $\lambda(E[h]-1)/(N-1)$. Any node in the ad-hoc network may be a relay for $N-1$ other nodes. Therefore, the expected amount of relay traffic at any node is: $\lambda(E[h]-1)$. The average total traffic per node, Λ, is the sum of the node's self traffic, λ, and all relay traffic reaches that node:

$$\Lambda = \lambda + \lambda(E[h]-1) = \lambda E[h] \tag{2}$$

where $E[h]$ is the excepted value of the hop count.

For correct reception of radio signals, the Carrier to Interference ratio (C/I) needs to be higher than a certain threshold value. C/I is the ratio between the mean power of wanted signal and the mean power of the sum of interfering signals. In radio communications the capacity of the networks is directly linked to the expected value of C/I. If we know the expected value of C/I, we can use the Shannon channel capacity formula to find an upper bound on the reliable data transmission speed between two nodes over the radio channel:

$$W = B\log_2(1 + E[C/I]) \tag{3}$$

Here B is the channel bandwidth[2] in Hz, $E[C/I]$ is the expected carrier to interference ratio, and W is the maximum capacity of the wireless radio channel. According to [24], the expected value of carrier to interference ratio in a MANET, $E[C/I]$, depends on the total number of nodes N, the reach of nodes in the center of the configuration α, path-loss exponent η, processing gain g, and the probability of transmission per node, the relational expression can be given by:

$$E[C/I] = \frac{g\sum_{j=1}^{\alpha} j^{-(\eta-1)}}{3\alpha(\alpha+1)^{-(\eta-1)}(1-e^{-\lambda E[h]})\sum_{j=1}^{\lfloor k/(\alpha+1)\rfloor} j^{-(\eta-1)}} \tag{4}$$

Here, λ is the mean arrival rate of new packets per node per time-slot (node's self traffic) and $E[h]$ is the average number of hops in MANET. In the study of wireless communications, the value of pathloss exponent is normally in the range of 2 to 4 (where 2 is for propagation in free space, 4 is for relatively lossy environments and for the case of full specular reflection from the earth surface); in this paper, because we consider the two adjacent nodes, the relative distance is small enough, the value of the pathloss exponent in this equation is 2, $\eta = 2$. In 802.11, the processing gain is realized by modulating each data bit with an 11 bit Barker code (pseudo-random sequence). Processing gain g is therefore 11:1, or 10.4 dB [26]. In [24], the reach of nodes in the center of the configuration α, the relationship between the total number of the network N and the size of the network k, and the average number of hops in MANET $E[h]$ have been discussed. According to the results in [24], we get:

$$\alpha = 1; E[h] \simeq 0.53\sqrt{N}; \tag{5}$$

$$k = \left\lfloor \sqrt{\frac{1}{4} + \frac{N-1}{3}} - \frac{1}{2} \right\rfloor \tag{6}$$

where the sign $\lceil x \rceil$ indicates rounding up to the nearest integer.

Finally, by introducing these values into the Eq. (4), the estimated equation, which is tailored for our clustering algorithm, can be given as:

$$E[C/I] = \frac{2g}{3\left(1 - e^{-\lambda E[h]}\right) \sum_{j=1}^{\lfloor k/2 \rfloor} j^{-1}} \tag{7}$$

where the sign $\lfloor x \rfloor$ indicates rounding down to the nearest integer.

In ad hoc networks, an additional restriction on the capacity is imposed by the MAC protocol [12], [15], [24]. Whenever a transmission link is established between two nodes, a portion of other nodes in the network will be prohibited from simultaneous transmission, because all these nodes are sharing the same transmission medium. Under fair conditions, the capacity of the radio channel is equally divided between all nodes competing to gain access to the medium. In the basic form of CSMA/CA at any moment in time only one of the neighboring nodes may transmit. Assume that the neighboring node degree of a node is n, the channel capacity needs to be divided by $n + 1$ to obtain the capacity, $R_{\max}$, per node:

$$R_{\max} = \frac{B}{n+1} \log_2 (1 + E[C/I]) \tag{8}$$

Here R_{max} is in bits per second and indicates the upper bound on the time-averaged error free bit transmission speed per node. If traffic conditions are such that the output bit rate per node of the source R_{out} tends to exceed $R_{\max}$ (i.e. $R_{out} > R_{\max}$), the network has capacity problems.

Based on [25], we can find the relation between the input bit rate per node R_{in}, and the output bit rate per node R_{out}. However, for translation from packets per time-slot to bits per second we need the exact duration of a time-slot t_s, the amount of overhead within each time slot t_{ov} and the useful data transmission interval t_d, and $t_s = t_{ov} + t_d$. The overhead time is the time needed for transmission of preamble and header in each data frame. Further, the overhead time includes the required inter-frame spacing times and the required time for the reception of MAC Acknowledgments for each data frame. A typical value for t_{ov} in IEEE 802.11b is 364 μs [26]. The length of t_d depends on data packet size P, and data transmission speed r, then $t_d = P/r$. In IEEE 802.11b, P may vary between 34 to 2346 bytes, while r is either 1 Mbps, 2 Mbps, 5.5 Mbps or 11 Mbps [26]. The input bit rate per node R_{in}, and the output bit rate per node R_{out}, relate to λ and Λ as:

$$R_{in} = \frac{\lambda P}{t_s} \tag{9}$$

$$R_{out} = \frac{\Lambda P}{t_d} = \frac{\lambda P E[h]}{t_d} \tag{10}$$

To avoid the capacity problem, the output bit rate per node of the source R_{out} cannot exceed $R_{\max}$, $R_{out} \le R_{\max}$, we get the upper bound of the number of CHs n:

$$\frac{\lambda P E[h]}{t_d} \le \frac{B}{n+1} \log_2 (1 + E[C/I]) \tag{11}$$

Finally, we get:

$$n^* = \frac{B \log_2 (1 + E[C/I])}{\lambda r E[h]} - 1 \tag{12}$$

By introducing the total number of nodes in a MANET N into the Eqs. (5) and (6), we can easily get $E[h]$ and k, then we get $E[C/I]$ using Eq. (7). Thus, as a matter of fact, the Eq. (12) is a function of N, and the value of n^* uniquely depends on the value of N.

3.1.2 Average Relative Distance θ

In existing wireless ad hoc network, the distance between the two mobile neighbor nodes is practically observable. Such information is inherent in the RSSI (or received signal strength indication) of a reachable mobile node. Measured at the node, RSSI can be modeled as the sum of two terms: one due to path loss, and another due to shadow fading.

Wireless transmissions are severely impaired by the multipath propagation effect. When a receiver receives multiple attenuated and time-delayed versions of a transmitted signal, with the additional corruption by noise and interference, the transmitted signal might be enhanced, thereby translating into the increasing signal to interference plus noise ratio (SINR), or weakened, thereby translating into the decreasing SINR. This is called multi-path fading and can be further divided into large-scale fading and small-scale fading. The shadowing phenomena considers the case that the received signal strength may be different due to the different propagation conditions in their surroundings even though the distance between two transmitter-receiver pairs is the same. Hence, it is referred to as large-scale fading.

We assume the radio propagation model:

(1) All nodes are with the same transmission power P_t.

(2) The attenuation is with distance and shadow fading.

(3) The shadow-fade attenuations between all pairs of source and destination nodes are independent and identically distributed (*i.i.d.*).

(4) The shadow-fade attenuation $10^{Z_{ij}}/10$ between any two nodes i and j is log-normally distributed and is the same, regardless of which node is the transmitter and which is the receiver. Thus, we have the pdf of the log-normal shadowing variable z is:

$$f_Z(z) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{z^2}{2\sigma^2}\right) \tag{13}$$

where σ is the same for all node pairs (i, j), and is the log-normal spread, i.e., the standard deviation of the Gaussian distribution that describes the shadowing phenomenon.

A node i has a connection to (i.e., is one hop away from) node j if and only if the received power P_r at node j exceeds some given threshold P_{min}, i.e., if and only if:

$$K\frac{P_t}{d_{ij}^{\delta}}10^{z_{ij}/10} = P_r \geq P_{\min}$$
$$\Leftrightarrow d_{ij} \leq \left(K\frac{P_t}{P_{\min}}\right)^{1/\delta}\exp\left(\frac{z_{ij}\ln 10}{10\delta}\right)$$
$$= H\exp\left(hz_{ij}/\delta\right) \quad (14)$$

Thus, the relative distance $RD_j(i)$ between the node i and j can be measured by:

$$RD_j(i) = \left(K\frac{P_t}{P_r}\right)^{1/\delta}\exp\left(\frac{z_{ij}\ln 10}{10\delta}\right)$$
$$= \left(K\frac{P_t}{P_r}\right)^{1/\delta}\exp\left(\frac{hz_{ij}}{\delta}\right) \quad (15)$$

where δ is the distance-loss exponent, $10^{z_{ij}}/10$ is the shadow fade between nodes i and j, d_{ij} is the distance between the nodes, K is a constant, taking into account parameters like the antenna gain, the antenna height (again assumed to be equal for all nodes), etc., $H = (P_tK/P_{min})^{1/\delta}$ is the range in the absence of fading, and $h = ln10/10$. For the common parameter values, $\sigma = 4 - 8$ dB, $K = 10$, $\delta = 3.5$.

In addition to measuring distances to neighbors, every node also piggybacks its collection of distance estimates to the periodically broadcasted "Msg_Node_Hello" messages so that the information is disseminated to all of its one-hop neighbors. The distance measurement is carried out by every node at a frequency of $1/\Delta t$ (the time interval Δt is equal to the time interval between two consecutive and successive "Msg_Node_Hello").

The average relative distance θ at node j is calculated by the variance of the entire set of relative distance values $RD_j(X_i)$, where X_i ($i \in [1, 2, \ldots, m]$) is a neighbor of j:

$$\theta = var\left(RD_j(X_1), RD_j(X_2), \ldots, RD_j(X_m)\right)$$
$$= E\left[\left(RD_j\right)^2\right] \quad (16)$$

Here *var*'s meaning is similar to that in Sect. 3.1.2, and is equal to $E\left[\left(RD_j\right)^2\right]$ which is the expected value of the squares of the m relative distance samples from j's neighbors.

3.1.3 Average Relative Speed ξ

Recently, a proposed method makes use of the time-varying inter-node range information for velocity estimations provided that the range estimates are noise-free. In this paper, a novel range-based method for relative velocity estimations (RVE) [33] is applied to obtain the relative speed between any two neighbor nodes. In addition to being less dependent on the characteristics of the wireless channel, RVE method is more tolerant of the multi-path or non-line-of-sight (NLOS) errors contained in distance measurements. Thus, this method is more robust than the many other algorithms in noisy communication environments.

Based on the discussions in [33], as shown in Fig. 1,

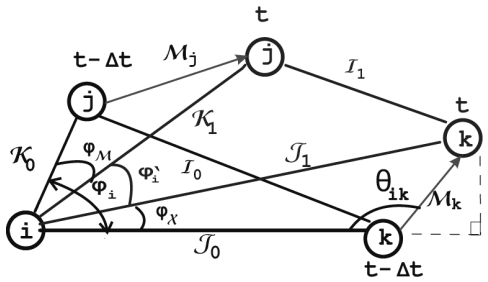


Fig. 1 Relative speed caused by their relative movements during time slot Δt.

suppose that a given node i has a neighbor node k. The node k's speed relative to node i during time slot Δt, $\widehat{v_k}$ by:

$$RM_j(i) = \widehat{v_k} = \frac{\sqrt{\left(J_1\cos\varphi_\chi - J_0\right)^2 + \left(J_1\sin\varphi_\chi\right)^2}}{\Delta t} \quad (17)$$

And the relative orientation $\widehat{\theta_k}$ can be estimated as:

$$\widehat{\theta_k} = \cos^{-1}\left(\frac{J_0^2 + \left(J_1\cos\varphi_\chi - J_0\right)^2 + \left(J_1\sin\varphi_\chi\right)^2 - J_1^2}{2J_0\sqrt{\left(J_1\cos\varphi_\chi - J_0\right)^2 + \left(J_1\sin\varphi_\chi\right)^2}}\right) \quad (18)$$

The average relative speed ξ at node j is calculated by the variance of the entire set of relative speed values $RM_j(X_i)$, where X_i ($i \in [1, 2, \ldots, m]$) is a neighbor of j:

$$\xi = var\left(RM_j(X_1), RM_j(X_2), \ldots, RM_j(X_m)\right)$$
$$= E\left[\left(RM_j\right)^2\right] \quad (19)$$

Here, *var* denotes the variance with respect to zero (and not the mean of the samples) and is equal to $E\left[\left(RM_j\right)^2\right]$ which is the expected value of the squares of the m relative speed samples from j's neighbors.

3.1.4 Average Residual Energy ε

After collected the neighbors' information, node j knows the residual battery power of each of its 1-hop neighbors. Let the residual battery power of node j with respect to its neighbor i be defined by $RE_j(i)$, and the entire set of neighbor nodes' residual battery power values be defined by $RE_j(X_i)$, where X_i ($i \in [1, 2, \ldots, m]$) is a neighbor of j, thus the average residual energy ε is calculated by:

$$\varepsilon = \frac{1}{m}\sum_{i=1}^{m} RE_j(X_i) \quad (20)$$

3.1.5 Average Link Stability μ

We define the link stability *LN* between two neighbor nodes as a prediction of the time for which the nodes will be within communication range of each other. The value of *LN* indicates that how long any particular node pair offers a stable link. The approach we adopt to estimate the link stability

LN is described as follows:

(1) Assume that node i broadcasts the message containing its location information, node j receives the last second packet at time $t-1$ and the last one at time t. Extracted from the received messages, j may obtain the location of a neighbor node i easily, denoted as (X_{t-1}, Y_{t-1}) and (X_t, Y_t).

(2) Node j judges whether node i is moving away or coming close. If $RSS^0_{i\to j} < RSS^1_{i\to j}$, then $RM_j(i) < 0$, and a negative value of the metric indicates that the two nodes are moving away. On the other hand, if $RSS^0_{i\to j} > RSS^1_{i\to j}$, then $RM_j(i) > 0$, and it indicates that the nodes are moving closer to each other.

(3) Node j estimates the link stability $LN_j(i)$:

If node i is moving away, the link stability between node pair i and j, $LN_j(i)$, is calculated as follows:

$$\begin{aligned} LN_j(i) &= \frac{D}{v} \\ &= \Delta t \times \left(R - \sqrt{(x_t - X_t)^2 + (y_t - Y_t)^2}\right) \\ &\quad \times \left| \sqrt{(x_t - X_t)^2 + (y_t - Y_t)^2} \right. \\ &\quad \left. - \sqrt{(x_{t-1} - X_{t-1})^2 + (y_{t-1} - Y_{t-1})^2} \right|^{-1} \end{aligned} \tag{21}$$

where D is the distance being remained before i is out of transmission range of j; v is the relative velocity between two nodes, because Δt, which is the time interval between t and $t-1$, is so small that we consider the velocity of any node is constant. Let denote (x_{t-1}, y_{t-1}) and (x_t, y_t) be j's location at time $t-1$ or t, respectively.

If i is moving towards j, then:

$$\begin{aligned} LN_j(i) &= \frac{D}{v} \\ &= \Delta t \times \left(R + \sqrt{(x_t - X_t)^2 + (y_t - Y_t)^2}\right) \\ &\quad \times \left| \sqrt{(x_t - X_t)^2 + (y_t - Y_t)^2} \right. \\ &\quad \left. - \sqrt{(x_{t-1} - X_{t-1})^2 + (y_{t-1} - Y_{t-1})^2} \right|^{-1} \end{aligned} \tag{22}$$

Assume that the multi-hop link from node j to node i is through relay node 1, node 2,..., node n, the link stability $LN_j(i)$ is defined as:

$$\begin{aligned} LN_j(i) &= LN_j(1) \cdot LN_1(2) \ldots \cdot LN_n(i) \\ &= \prod_{(a,b)=\{(j,1),\ldots,(n,i)\}} LN_a(b) \end{aligned} \tag{23}$$

Similar with the ε, let the link stability of node j with respect to its neighbor i be defined by $LN_j(i)$, and the entire set of neighbor nodes' link stability values be defined by $LN_j(X_i)$, where X_i ($i \in [1, 2, \ldots, m]$) is a neighbor of j, thus the average link stability μ is calculated by:

$$\mu = \frac{1}{m} \sum_{i=1}^{m} LN_j(X_i) \tag{24}$$

3.2 Clustering Procedure

In this section, we present our weight based clustering algorithm. Clustering can be modeled as a network partitioning problem with some added constraints. We first give the basis for our algorithm and then discuss the details.

3.2.1 Basis for Our Algorithm

To decide how well suited a node is for being a CH, we take into account its node degree, relative speed, relative distance, residual battery power and link stability. The following features are considered in our clustering algorithm:

- The CH election procedure is not *periodic* and is invoked as rarely as possible. This reduces system updates and hence computation and communication costs. The clustering algorithm is not invoked if the relative distances between the nodes and their CHs do not change. The CH election procedure is invoked at the time of system activation and also when the current CHs set is unable to cover all the nodes. Every invocation of the election algorithm does not necessarily mean that all the CHs in the previous CHs set are replaced with the new ones. If a node detaches itself from its current CH and attaches to another CH, then the involved CHs update their member list instead of invoking the election algorithm.
- We assume that a node always knows the number of nodes N in the MANET. Each CH can reasonably support only $\delta = N/n^*$ (As mentioned in Sect. 3.1.1, by introducing N, we can calculate n^* and then get δ. We consider that δ is a pre-defined threshold and setup it as default before the MANET deployed because we assume that every node knows the total number of nodes in MANET) nodes to ensure efficient MAC functioning. If the CH tries to serve more nodes than it is capable of, the system efficiency suffers in the sense that the nodes will incur more delay and the wireless channel may have more packet collision. A high system throughput can be achieved by limiting or optimizing the degree of each CH. To any node i, the degree of the node, Deg_i, is defined as the total number of 1-hop neighbors (nodes within a node's maximum communication range):
  $Deg_i = \{x |\ x$ is a node within the maximum communication range of $i\}$
  Then the *node degree difference* of i, Δ_i, is:
  $$\Delta_i = |Deg_i - \delta| \tag{25}$$
- A CH is able to communicate better with its neighbors having closer *relative distances* from it within the smaller communication range [1], [15]. As the nodes move away from the CH, the communication may become difficult due mainly to signal attenuation with increasing distance.

- *Mobility* is an important factor in deciding the CHs. In order to avoid frequent CH changes, it is desirable to elect a CH that does not move away from its CMs very quickly. When the CH moves fast, the nodes may be detached from the CH and as a result, a re-affiliation occurs. Re-affiliation takes place when one of the CMs moves out of a cluster and joins another existing cluster. In this case, the amount of information exchange between the node and the corresponding CHs is local and relatively small. However, when the current CH set is unable to cover all the nodes, the re-clustering occurs. Because all the nodes are participating in this process, the information update in the event of the change in re-clustering is much more than a re-affiliation.
- The *battery power* can be efficiently used within certain transmission range, i.e., it will take less power for a node to communicate with other nodes if they are within close distance to each other. A CH always consumes more battery power than a CM since a CH has extra responsibilities to carry out for its members.
- *Link stability*, which has been defined in Sect. 3.1.5, is another important factor. A higher stability indicates that the route between the nodes can persist for a certain time span, which means that the re-affiliation procedure may occur at reasonably low frequency.

3.2.2 Proposed Algorithm

Based on the preceding discussions, we propose a novel algorithm that effectively combines each of the above parameters with certain weight factors chosen according to the application needs. The flexibility of changing the weight factors helps us apply our algorithm to various given missions. Our algorithm is divided into three steps. The first step performs a simple neighbor discovery protocol and assigns a weight for each node. The second step elects a set of CHs and then formats the cluster. The third step connects the CHs together forming a connected virtual backbone. In the following, each step is described and analyzed.

Step 1: Neighbor discovery and weight generation

All nodes send and receive "Msg_Node_Hello" messages to/from their 1-hop neighbors, which contains <ID, GPS(x,y), RE>. To any node i, it gets the *RSS* of two consecutive and successive "Msg_Node_Hello" messages from every neighbor, and then obtains its degree difference Δ_i using Eq. (25), average relative speed ξ_i using Eq. (19), average relative distance θ_i using Eq. (16), average residual energy ε_i using Eq. (20), and average link stability μ_i using Eq. (24). At the meanwhile, i estimates its residual energy RE_i which implies how much battery power left. Then i compares its RE_i with ε_i, if $RE_i \geq \varepsilon_i$, i is qualified to be a CH candidate; otherwise, i gives up the opportunity in this process.

Our algorithm tends to give a high weight for nodes that have: (1) low degree difference, (2) low average relative speed, (3) low average relative distance, (4) high residual energy, (5) high average link stability. According to these features, the combined weight W_i for any node i can be computed as:

$$W_i = w_1\Delta_i + w_2\xi_i + w_3\theta_i + w_4\varepsilon_i + w_5\mu_i \tag{26}$$

where w_1, w_2, w_3, w_4 and w_5 are the weight factors for the corresponding system parameters.

Step 1: Analysis

Step 1 requires each node to send 2 messages ($O(1)$ message complexity). The two messages are used to send the node's initial information <ID, GPS(x,y), RE>. The time complexity is $O(2)$ because each node needs to receive the messages from its neighbors.

Step 2: Elect CHs and format cluster

All nodes start in "Cluster_Undecided" state. After generated the node weight, each qualified node broadcasts its own combined weight in "Msg_Node_Weight" <ID, GPS(x,y), W>, and the other nodes send a message "Msg_Node_Info" <ID, GPS(x,y)> to the neighbors. Any node receives the message from its neighbors, stores the information in a special data structure called neighbor table *NT* (vector), and then compares its own combined weight W with those of the neighbor table. If a node has a highest value of W amongst all its neighbors, it assumes the status of "Cluster_Head_Candidate" and then declares itself to be as a CH candidate node; otherwise it declares itself to be a "Cluster_Member" and then waits to join a CH. If two neighboring CH candidate nodes in "Cluster_undecided" state have the same value of W, we resort to comparison of average relative mobility. The node whose ξ is smaller is decided to be as the CH, and changes its status to be "Cluster_Head"; the loser has to change its status to be "Cluster_Member". Furthermore, because in a mobile scenario, if two nodes with status "Cluster_Head" move into each other's communication range, and they are in contention of retaining the "Cluster_Head" position, the "Event_Retain_CH" occurrence is deferred for a "CH_Contention_Interval" (i.e., 2× ACK Timeout Interval [26]) to allow for the incidental contacts between passing nodes. If the nodes are in communication range of each other even after the "CH_Contention_Interval" timer has expired, the "Event_Retain_CH" occurs; the node with the smaller ξ keeps the status of "Cluster_Head", and the other changes its status to be "Cluster_Member".

Step 2: Analysis

In step 2, every node sends 1 message (to declare its status) and little portion of these nodes send another message (to challenge the CH position) ($O(1)$ message complexity). If a node has the maximum weight among it neighbors, it sends a message informing them that it declared itself as a

CH. If the node does not have the maximum weight among its neighbors, it broadcasts a "Cluster_Member" message.

Let $Deg_{\max}$ be the maximum node degree (i.e. the maximum number of neighbors of any node) in the MANET. The time complexity of this step is $O(Deg_{\max})$ because a node needs to compare its own combined weight W with those of the neighbor table NT. If the 1-hop neighbors are sorted according to their weight, the time complexity of this step becomes $O(Deg_{\max} \log(Deg_{\max}))$.

Theorem 1: Let $h(i, j) = k$ represent that there are k hops between i and j. To any CM i, there exists a CH j such that $h(i, j) = 1$.

Proof. The role of any node is as either a CH or a neighbor of a CH. As mentioned above, a node has a highest value of W amongst all its neighbors, it will be a CH; otherwise, it will be a CM.

Theorem 2: To any CH m, it can reach its neighbor CH n in 2 or 3 hops such that $h(m, n) = 2$ or $h(m, n) = 3$, and the intermediate hops are CMs called "Gateway node" (GW).

Proof. According to the specifications in Step 2, when a CH m from its cluster moves into another cluster, after the "CH_Contention_Interval" timer has expired, it challenges the corresponding CH n, and then one of them will give up its CH position. This rule means that $h(m, n) \neq 1$. Let $C_y(x)$ ={x| x is a neighbor CH of node y}. Assume that a node i and a CH m, if $h(i, m) = 1$, then i is a CM. If exits a node n, $\exists n \in C_i(n)$ and $n \notin C_m(n)$, then $h(m, n) = 2$. If $n \notin C_i(n)$ and $n \notin C_m(n)$, then n is a CM. But according to Theorem 1, node n must have at least one CH neighbor. Let such a CH node be j, then $h(m, j) = 3$. Therefore, any CH node can reach its neighbor CH in 2 or 3 hops.

Based on the analysis above, to be simple and clear, here are some definitions as follows:

The mobile ad hoc network formed by the mobile nodes and the links can be represented by an undirected graph $G = (V, E)$, where a node set $V = \{v\}$ and a node connectivity set $E = \{e\}$.

Definition 1: There are two CHs: j and k, and a CM m. Assume that m is a member node of j, denoted as $m \in j$; meanwhile, m is not a member node of k, denoted as $m \notin k$. If the links $(m, j) \in E$, $(m, k) \in E$ exist, the CM m is defined as cross node (CN) for CH j and CH k.

Definition 2: There are two CHs: j and k, and two CMs: m and n. If $(m \in j) \wedge (n \in k)$, and the links $(j, m) \in E$, $(m, n) \in E$, $(n, k) \in E$ exist, the CM m and CM n are defined as joint nodes (JN) of CH j and CH k.

Definition 3: Gateway node (GW) consists of two types: CN and JN.

Step 3: Connect CHs and construct the backbone

According to Theorem 2, each CH is 2-hops or 3-hops away from its neighbor CH. To connect the CHs, every CH sends a message "Msg_Connect_CH" <CH_ID, GPS(x,y), h> to its CMs. Specially, the default value of h is set 3 and decreases by 1 when the message transmitted through a node; when $h = 0$, if the node received the message is not a CH, the message will be dropped. Moreover, when relaying the message, the node adds its information <ID, GPS(x,y)> into the message. The CM, whose role is GW, receives and relays the message to its next hop; otherwise, the message is rejected. If the next hop is a CM, who is not GW, the message is dropped. If the next hop is CH, it receives the message, stops to relay the message but stores it in its neighbor CH table NHT (vector).

This process connects a CH to other CH that is 2-hops and 3-hops away from itself, where the intermediate hops are GWs. After that, the connected CHs set is formed. Each CH should select some GWs to forward the packet when it sends the packet to all the neighbor CHs. Thus, the connected CHs set consists of all the CHs and GWs.

In this work, we use the algorithm, presented by Orhan Dagdeviren and Kayhan Erciyes [28], to construct a virtual backbone architecture on the clustered MANET using the connected CHs set. Different from other algorithms, the virtual backbone is constructed as a directed ring architecture to gain the advantage of the cluster topology and to give better services to other middleware protocols, for instance, our C-LAR protocol in this paper. The backbone formation algorithm [28] only focused on that-how to construct the backbone over a clustered MANET using the directed ring architecture. At the initialization stage, authors assumed that the MANET has been partitioned by a clustering algorithm. And then, the backbone formation algorithm was discussed. However, the CHs election, the cluster formation and the nodes connection, which are key issues related to the clustering algorithm, had not been mentioned and discussed. Different from the backbone formation algorithm, C-LAR firstly partitions the nodes into clusters, each with a CH and some CM nodes, so that the CHs form an independent set. Step 1 indicates that the choice of the CH is performed based on a combined "weight" associated to a node. This attribute basically expresses how fit that node is to become a CH. Step 2 describes that how to elect and produce a set of CHs that are independent, and the criteria for joining them to form a connected backbone have been defined. The role we adopted has been proved by Theorem 2. Finally, the CHs and gateways form the backbone node set, which is the fundamental source for applying the backbone formation algorithm [28].

The main idea of the backbone formation algorithm is to maintain a directed ring architecture by constructing a minimum spanning tree between CHs and classifying CHs into *BACKBONE* and *LEAF* nodes, periodically. After clustered the MANET, each node has known its CH's ID. To maintain these structures, each CH broadcasts a CH_Info message. In this phase, CM nodes act as routers to transmit these CH_Info messages. This algorithm uses the hop-based backbone formation scheme. According to Theorem

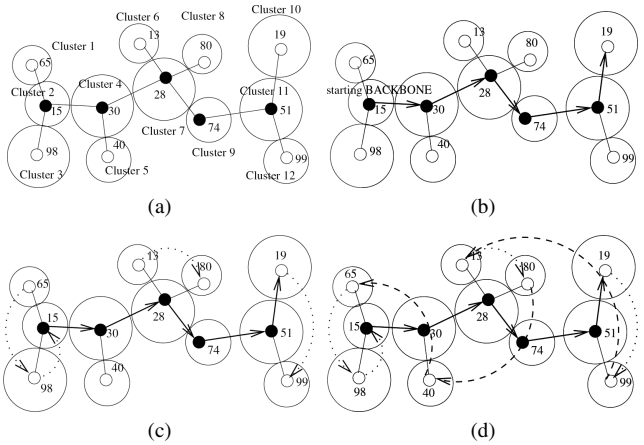


Fig. 2 An example operation for backbone formation.

2, any CH reaches its neighbor CH in 2 or 3 hops, and the values are taken into consideration in a minimum spanning tree construction. *BACKBONE* CHs are shown as black and *LEAF* CHs are shown as white nodes. The main part of the algorithm is the construction of a ring architecture by orienting CHs in the minimum spanning tree. General idea is to divide the ring into two parts: a directed path of *BACKBONE* CHs and a directed path of *LEAF* nodes. Finally, these two directed paths are connected to each other to maintain the ring architecture. Each CH aims to find the next CH to construct the ring architecture by the procedure "RING_Construct", which is described in [28].

The first aim is to form the vital part of the backbone. The *BACKBONE* CHs are directed to each other from starting *BACKBONE* CH to the end. Starting *BACKBONE* CH is the one with the smallest connectivity to other *BACKBONE* nodes. This selection policy of *BACKBONE* CH results in smaller hops and reduced routing delay. Ending *BACKBONE* CH is directed to its *LEAF* CHs. *LEAF* CHs firstly execute the procedure "Ordinary_LEAF_Proc" to find the next CH on the ring. The aim of directing *LEAF* CHs with the same *BACKBONE* CHs to each other is to make the routing process over the same *BACKBONE* CH to reduce delay. *LEAF* CHs which can't find the next CH execute the procedure "BACKBONE_Proc" and search for a *LEAF* CH from the previous *BACKBONE* CHs of their parent to find a *LEAF* CH. The last aim is to connect the *LEAF* CHs of different *BACKBONE* parents to maintain the routing operation by using the *BACKBONE* CHs.

Here an example operation is given to show the backbone formation. Assume the MANET with CHs in Fig. 2 (a). Clusters have been formatted using the C-LAR. Nodes 65, 15, 98, 30, 40, 13, 28, 80, 74, 19, 51 and 99 are the CHs of clusters 1 to 12, respectively. Each CH broadcasts the CH_Info message to its neighbors. After each CH receives the CH_Info message of the others, minimum spanning tree in Fig. 2 (a) is constructed by all CHs. Nodes 65, 98, 40, 13, 80, 19 and 99 identify themselves as *LEAF* CHs, while nodes 15, 30, 28, 74 and 51 identify themselves as *BACKBONE* CHs. *BACKBONE* CHs are filled with black and *LEAF* CHs are filled with white as shown in Fig. 2 (a).

To connect the *BACKBONE* nodes, a starting *BACKBONE* CH must be chosen. The criteria are to select the *BACKBONE* node which has the smallest connection to other *BACKBONE* CHs. Node 15 is connected to 30, 30 is connected to 15 and 28, 28 is connected to node 30 and node 74, node 74 is connected to node 28 and 51, 51 is connected to 74. Node 15 and 51 can be the choice for starting *BACKBONE* CH. 15 is selected because its ID is smaller than 51. 15 selects the next CH as 30, 30 selects the next CH 28, operation continues in this way. The ending *BACKBONE* CH directs to its *LEAF* with the smallest node id. These directions can be seen in Fig. 2 (b) with bold directed lines. *LEAF* CHs of a *BACKBONE* CH are directed to each other from smallest to greatest. Node 19 is directed to 99, 13 is directed to 80, 65 is directed to 98 as seen in Fig. 2 (c) with dotted directed lines.

Lastly, *LEAF* CHs of different *BACKBONE* CHs are connected as in Fig. 2 (d). Each *LEAF* leader which can not find the next CH, searches for a *LEAF* CH from the children of the previous *BACKBONE* CH of its parent *BACKBONE* CH. Node 99 is connected to 13, node 80 is connected to 40, node 40 is connected to 65, 98 is connected to 15 shown with dashed lines in Fig. 2 (d).

A virtual backbone plays a key role in routing as it simplifies the routing process to one in a smaller sub-network from the connected CHs set. Using the virtual backbone nodes, routing messages are mainly exchanged between the backbone nodes, instead of being broadcasted to all the nodes. Thus, the appropriate virtual backbone from the current connected CHs set is able to reduce the routing overhead, to minimize the routing delay and to simplify the connectivity management.

Step 3: Analysis

In step 3, the entire procedure is divided into two parts: connect the CHs and construct the backbone. In the beginning of connecting the CHs, each CH node sends one message. When a CM, whose role is GW, receives it, the node adds its local information and re-sends it to the next hop node. The upper bound is therefore $O(1)$. Using the virtual backbone scheme, assume that we have n CHs in the network. Every node is connected to its CH and the entire CHs set is connected with each other. n CHs have to flood the message to the network. Total number of message in this case is Nn which means that the complexity has an upper bound of $O(N)$. As a result, the message complexity of this procedure is $O(N + 1)$.

In a similar manner, the time complexity of connecting the CHs is $O(Deg^2_{max})$ because a CH needs to loop across its 1-hop and 2-hop neighbors. Note this step is only executed by the CHs. Besides, on constructing the virtual backbone, flooding of m messages to the network takes Nm time. Thus, the time complexity is at most $O(Nm + Deg^2_{max})$.

4. C-LAR Protocol

In this section, we describe the Cluster Based Location-Aided Routing Protocol (C-LAR) which runs on top of a cluster cover of the MANET. Firstly, some necessary definitions are given, including expected zone and request zone. Secondly, three scenarios of request zone are analyzed. Thirdly, route discovery, hole problem, and route recovery are discussed in detail.

4.1 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 with the LAR [10], we consider that, 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 participating in the routing process can be updated. In general, 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 experiments, we set the default maximum speed of node is 30 m/s, and that is known to all nodes.

4.2 Expected Zone

In the route discovery procedure, the source S uses the location information of the destination D to estimate the region that D expects to appear, the region is called as expected zone [10]. We extend the definition of expected zone in LAR, because many literatures have proved that the method proposed in LAR can not calculate the expected zone exactly [2], [3]. However, when the route request message arrived the original location of D, some time passed, this time interval can be called Δt. As shown in Fig. 3, in order to calculate a more exact expected zone, we must take the time interval Δt into account. There are two scenarios: (1) the routing path 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 path from S to D is not established. In this scenario, a challenge is how to determine the value of Δt. For simplicity, Δt can be set to half of the round trip time between S and D. Another more precise and complex method to get the value of Δt is as follows: when S received a packet from D, S adds the location information of D and the transmission time of this packet from D to S into its routing table. If S needs to calculate an expected zone for D, we can let Δt as the transmission time from D to S that is recorded in the routing table of S. Furthermore, node S can determine the expected zone based on the knowledge that node D at Location GPS(x_D, y_D) at time t_0. 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}. The Δt can be given as: $\Delta t = \frac{\sqrt{(x_S - x_D)^2 + (y_S - y_D)^2}}{v_{max}}$. Finally, the radius of estimated expected zone is $v_{max}[(t_1 - t_0) + \Delta t]$.

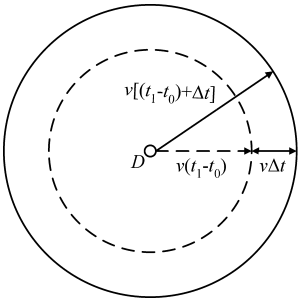


Fig. 3 Expected zone.

4.3 Request Zone

Instead of searching the route in the entire network blindly, C-LAR confines the route searching space into a smaller estimated region, which is defined as request zone [10]. A node forwards a route request only if it belongs to the request zone. To increase 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 it and the routing path. It is mainly due to the fact that there is no guarantee that a path can be found consisting only 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 with the hole problem in Sect. 4.5.4. In this event, however, the latency in determining the route to D will be longer (as more than one round of route discovery will be needed).

4.4 Selection of Request Zone

Generally, accuracy of request zone (i.e., probability of finding an available route to the destination) can be increased by increasing the size of request zone (i.e., the total number of nodes contained in this zone). Because the more number of nodes participates in the routing process, the greater the probability of establishing route path from S to D is, 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 [10], 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, we also consider that C-LAR 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 C-LAR, 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 same cluster; (Scenario III) S is within the expected zone. Assume that S is a node in a cluster which CH is H_1 (S may be H_1 when S is CH, in this case is denoted as S), D is a node in a cluster which CH is H_2 (D may be H_2 when D is CH, in this case is denoted as D). After expected zone has been estimated, we denote that $EZ[t] = v_{max}[(t_1 - t_0) + \Delta t]$ represents the expected zone at time t, and $d(S, D)[t]$ is the relative distance between S and D at time t. If $d(S, D)[t] \leq EZ[t] = v_{max}[(t_1 - t_0) + \Delta t]$, then S is in the $EZ[t]$, denoted as $S \in EZ[t]$; if $d(S, D)[t] > EZ[t]$, then S is out of the $EZ[t]$, denoted as $S \notin EZ[t]$. We define:

- [1]Scenario I: If $(S \in H_1 \text{ OR } S) \wedge (D \in H_2 \text{ OR } D) \wedge (H_1 \neq H_2) \wedge (S \notin EZ[t])$, then Scenario I is preferred;
- [2]Scenario II: If $(S \in H_1 \text{ OR } S) \wedge (D \in H_2 \text{ OR } D) \wedge (H_1 = H_2) \wedge (S \notin EZ[t])$, then Scenario II is preferred;
- [3]Scenario III: If $(S \in H_1 \text{ OR } S) \wedge (D \in H_2 \text{ OR } D) \wedge (S \in EZ[t])$, then Scenario III is preferred;

The criteria for scenario selection are described in the pseudo-code of Step 2 in Sect. 4.6.

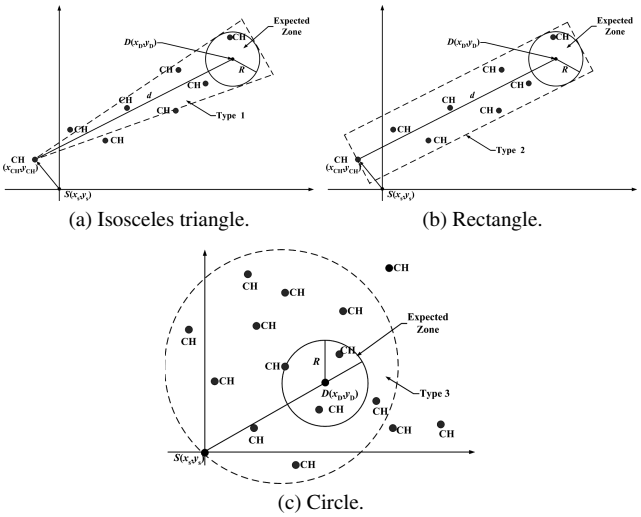
The types of request zones we introduced are listed as: the isosceles triangle, the rectangle, and the circle. To select the appropriate type of request zone according to the relation of relative location among S, D and EZ, we conduct a series of simulations, in which the configuration is the same as those in Sect. 5. To examine the performance, we introduce four metrics: (1) total times of collision, which we define as the total times of collision took place when using different types of request zone; (2) route set up time, which we define as the average time required to construct a path to D; (3) route discovery overhead, which we define as the total number of packets transmitted per node per route established from S to D; (4) probability of route recovery, which we define as the times of route recovery which is due to the link failure in each round and the denominator is the total times of route discovery.

4.4.1 Scenario I

In C-LAR, 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. When considering the network layer performance of Scenario I, the impact of this characteristic cannot be ignorable.

Figure 4 shows the cases when the request zone is defined as the isosceles triangle, the rectangle, or the circle respectively in Scenario I. From Fig. 5 (a), we observe that the number of collision in the case when the type of request zone defined as the isosceles triangle, is much less than the other twos, although the occurrence of collision for all the types of request zones raise as the node density increases. Generally, the probability of collision is proportional to the number of packets to be transmitted; and the more nodes needed to transmit packets will produce a mass scale of traffic and cause more collision. As above, the area of request zone defined as the isosceles triangle is the smallest of the three so that the smallest number of nodes participates in the route process. Intuitively, the less intermediate nodes involved, the less control packets are broadcasted. As a result, the total number of collision in the case when the type of request zone defined as the isosceles triangle is the smallest.

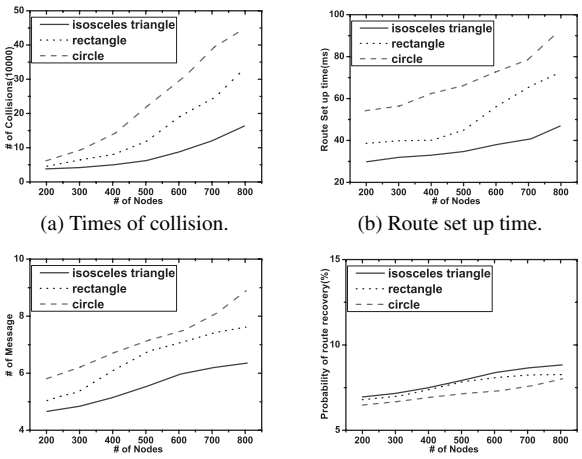
Figure 5 (b) shows that the average time, called route set up time, required to construct a route to the destination. The case when the type of request zone defined as the isosce-



(a) Isosceles triangle. (b) Rectangle.

(c) Circle.

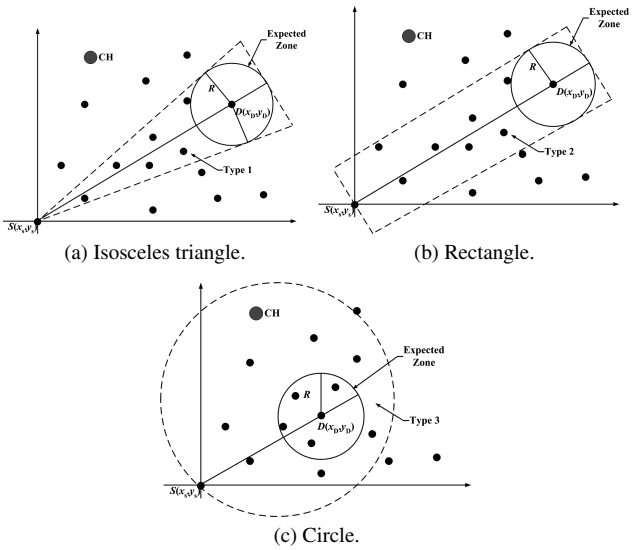
Fig. 4 Three types of request zones in Scenario I.



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

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

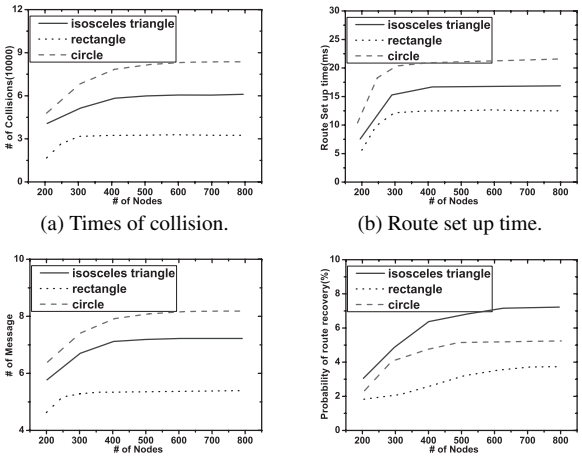
Fig. 5 Network layer performance comparison in RZI.



(a) Isosceles triangle. (b) Rectangle.

(c) Circle.

Fig. 6 Three types of request zones in Scenario II.



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

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

Fig. 7 Network layer performance comparison in RZII.

les triangle requires the less route set up time than the other twos for different network sizes. As the analysis above, in the case when request zone defined as the isosceles triangle, 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. On the other hand, the larger the number of packets is transmitted at the same time, the greater the chance of collision increases. Collision induces packet retransmission and lengthens transmission time. This results in a longer route set up time in the other two cases when the type of request zone defined as the rectangle or the circle.

Figures 5 (c) and (d) show that either the route discovery overhead or the probability of route recovery tends to initially increase for the node density scaling. We observed from Fig. 5 (d) that, the probability of route recovery in the case when the type of request zone defined as the isosceles triangle is only a bit worse than the other twos. As the analysis above, in the case when request zone defined as the isosceles triangle, it restrains the messages to forward along the narrowest space. It indicates that the isosceles triangle has the least number of nodes involved in the route process, the less nodes involved, the less packets are broadcasted, and the greater probability of route recovery is. Because the relative distance between S and D is quite large, 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.

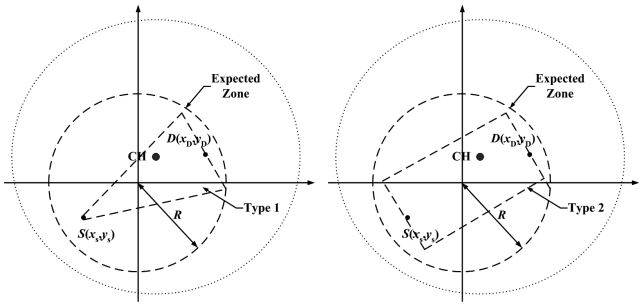
4.4.2 Scenario II

Figure 6 shows the cases when the request zone is defined as the isosceles triangle, the rectangle, or the circle respectively, in Scenario II. Note that the characteristic of scenario II, S is outside of expected zone, and S and D are in same cluster. C-LAR is implemented on a cluster cover. As mentioned in Sect. 3.2.1, each CH can reasonably support only a certain number nodes to ensure efficient MAC functioning. Thus, with the node density increasing, the number of CHs is also increasing, however, the number of 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.

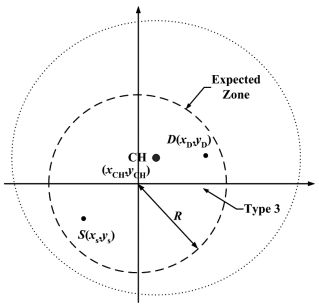
In Fig. 7, the results show that the four metrics increase gradually and then level off with the node density scaling; and meanwhile, in the case when the type of request zone defined as the rectangle, the four metrics are much better than those of the other twos. The area of the rectangle (i.e. the number of nodes in the area) is larger than that of the isosceles triangle, but smaller than that of the circle. Note that S and D are in a same cluster. Because of the mobile characteristic of nodes, too small area (i.e. few nodes) may incur a large probability of link failure. According to the simulations, the probability of route recovery in the case when the type of request zone defined as the isosceles triangle is always the largest. As mentioned above, 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 case when the type of request zone defined as the circle, the case when the type of request zone defined as the rectangle has smaller number of nodes involved in the routing process. As the analysis in Sect. 4.4.1, the less number of nodes participates in the routing process, the less the number of packets is transmitted simultaneously, the smaller the chance of collision increases and the smaller the discovery overhead is.

4.4.3 Scenario III

Figure 8 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

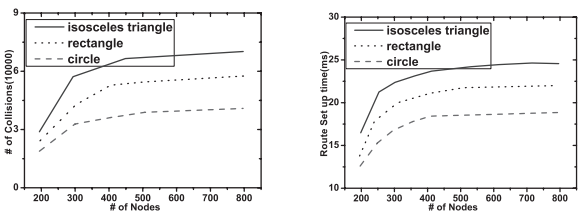


(a) Isosceles triangle. (b) Rectangle.



(c) Circle.

Fig. 8 Three types of request zones in Scenario III.



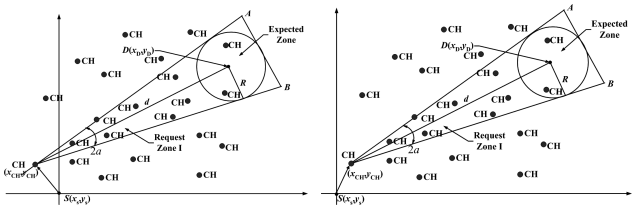
(a) Times of collision. (b) Route set up time.

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

Fig. 9 Network layer performance comparison in RZIII.

$v_{max}[(t_1 - t_0) + \Delta t]$. This characteristic indicates that the relative distance between S and D is quite small, which affects the four performance metrics deeply. Furthermore, as mentioned in Sect. 4.4.2, the limitation of the number of CMs in a cluster still has the influence.

We observe from Fig. 9 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 twos. It seems like that the simulation results are 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 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



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

Fig. 10 Request Zone I.

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 exists because of the narrow route discovery space and node movement, which is easier to make a link failure. When a link failure occurs in MANET, the algorithm tries search for new routes and re-establish the failure route. This behavior is detrimental to the network performance.

4.5 Scenarios of Request Zone

According to the analysis above, we define the form of request zone as the isosceles triangle, the rectangle, the circle corresponding to the Scenario I, Scenario II, Scenario III, respectively. The details of each scenario are discussed blow.

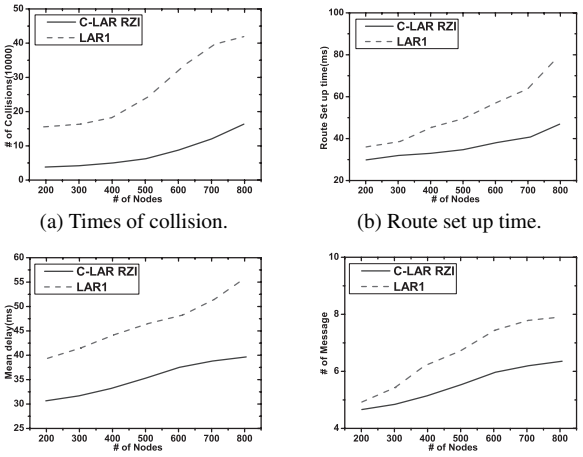
4.5.1 Scenario I

If S is outside of the expected zone, and S and D are in different clusters, the request zone is defined as a isosceles triangle, named RZI. As mentioned above, the virtual backbone from the connected CHs set 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". C-LAR 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 a higher chance to make the request message route 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.

Inspecting Fig. 10, the RZI corners are CH (whose location is GPS(X, Y)), A and B. The area of RZI can be calculated as follows:

$$\begin{aligned} S_\Delta &= (d+R)^2 \tan\alpha \\ &= \{v_{max}[(t_1 - t_0) + \Delta t]\} \\ &\quad \times \left(\sqrt{(X - x_D)^2 + (Y - y_D)^2} + v_{max}[(t_1 - t_0) + \Delta t] \right)^2 \\ &\quad \times \left\{ (X - x_D)^2 + (Y - y_D)^2 - (v_{max}[(t_1 - t_0) + \Delta t])^2 \right\}^{-1} \end{aligned} \tag{27}$$



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

(c) Mean delay. (d) Routing load.

Fig. 11 Network layer performance comparison in RZI.

In C-LAR, 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. Using the backbone nodes, the routing message is exchanged between the CHs, instead of being broadcasted to all the nodes. Thus, in the Scenario I, the algorithm always makes effort to make the routing path pass the CHs as many as it can.

Comparison with LAR1

In this section, we compare the network layer performance of the C-LAR with the well-known LAR scheme 1 (LAR1), when S is outside of the expected zone, and S and D are in different clusters.

We consider the four performance metrics: (1) total times of collision occurrence, which we define as the total times of collision took place when using different routing algorithms; (2) route set up time, which we define as the average time required to construct a path to D; (3) mean delay, which we define as the number of time steps required to deliver a data packet from S to D after the routing path has been built; (4) normalized routing load, which we define as the total number of packets transmitted per data packet delivered to D.

From Fig. 11 (a), we observe that the number of collision in C-LAR is much less than LAR1, although the occurrence of collision for both routing algorithms is raising with the node density scaling. Generally, the probability of collision is proportional to the number of packets to be transmitted. The more nodes needed to transmit packets will produce a mass scale of traffic and cause more collision. According to the results, the request zone defined by LAR1 is larger than that of C-LAR so that a greater amount of nodes takes part in the route probing. The more forwarding nodes participate in the routing process, the more control packets are broadcasted. This characteristic results in a higher chance of collision in LAR1 algorithm.

Figure 11 (b) shows that the average time, called route set up time, required to construct a path to a destination node for C-LAR and LAR1 algorithm. For both the routing algorithms, the route set up time increases when the network node density is growing. LAR1 requires longer route set up time than C-LAR for different network node densities. In C-LAR, the routing messages are transmitted through as many CHs as possible, the connected virtual backbone nodes which are from the CHs set play a main artery to exchange the messages, instead of flooding within the network. Furthermore, RZI zone restrains route request message to forward along a narrower space. It means that the message is force to propagate in as straight direction as possible. It is profitable to decrease the route set up time.

Figures 11 (c) and (d) show that the mean delay and the normalized routing load for the node density scaling. We observe that the mean delay values and the number of messages transmitted for a route increase as the node density increases for both routing algorithms. The results show that C-LAR has lower load and lower delay than LAR1. Because LAR1 defines a larger request zone than that in C-LAR and expands the request zone rapidly if last route discovery procedure fails, it induces a higher routing overhead. As can be seen, LAR1 produced a larger amount of control packets that caused a higher probability of collision. It also increases the route set up time (Fig. 11 (b)) and degrades the performance of data transmission (Fig. 11).

The simulation results demonstrate that C-LAR incurs a good network layer performance as the number of nodes in a fixed network area is scaled up for the inter-cluster route path. There are three basic advantages: (1) the expected carrier to interference ratio $E[C/I]$, (2) the connected virtual backbone, (3) the hierarchical structure. The metric $E[C/I]$ is used to determine radio channel capacity and useful data output rates per node. We find a good estimation for C/I so that it is propitious to the performance of packet collision and message transmission. The connected virtual backbone plays a key role in exchanging the messages between the CHs, instead of being flooded to all the nodes. The hierarchical structure can dynamically deal with the changes of network topology and improve reliability, thus it can decrease the probability of re-affiliation and the route recovery. Finally, the routing cost is decreased and the performance is improved.

4.5.2 Scenario II

If S is outside of the expected zone, and S and D are in same cluster, the request zone is defined as a rectangle, named RZII, which includes the current location of S and the estimated expected zone EZ. As shown in Fig. 12, the area of RZII, whose corners are S, A, B, C and E can be calculated as follows:

$$\begin{aligned} S_{ABCE} &= 2R \cdot (d + R) \\ &= 2R \cdot \left(\sqrt{(x_S - x_D)^2 + (y_S - y_D)^2} + R \right) \end{aligned} \quad (28)$$

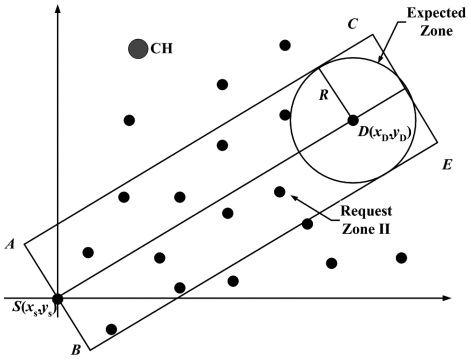
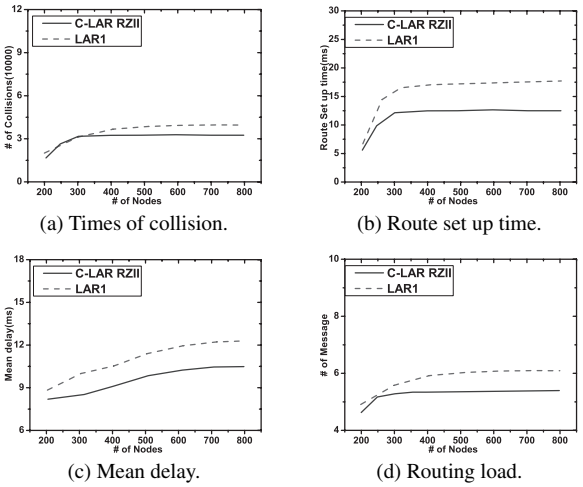


Fig. 12 Request Zone II.



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

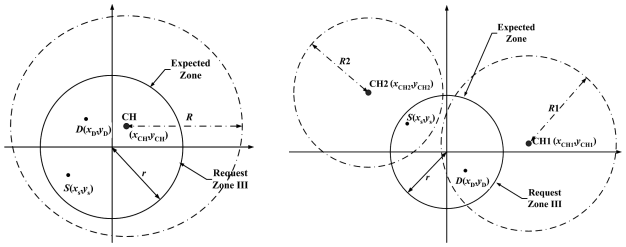
(c) Mean delay. (d) Routing load.

Fig. 13 Network layer performance comparison in RZII.

Comparison with LAR1

In this section, the same network configurations are used to compare the network layer performance of the C-LAR with LAR1, when S is outside of expected zone, and it is in same cluster with the destination D. As the analysis in Sect. 3.2.1, to ensure the cluster performance, each CH can handle a reasonable number of CMs. This characteristic affects the algorithm performance. As a result, each performance metric generally tends to initially increase and then levels off as the node density increases for both routing algorithms.

From Fig. 13, the results show that the four performance metrics in C-LAR are much better than those in LAR1. This is caused by the number of nodes participating in the routing process. When using the C-LAR, the area of request zone is much smaller than that in LAR1, it indicates that, in C-LAR, the number of nodes participating in the routing process is smaller than that in LAR1. Furthermore, similar with that in Scenario I, in C-LAR, the request message is also forced to propagate in as straight a direction as possible. This is preferable in providing a higher chance to select a shorter route. As observed in Sect. 4.5.1, it is also profitable to improve the performance.



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

Fig. 14 Request Zone III.

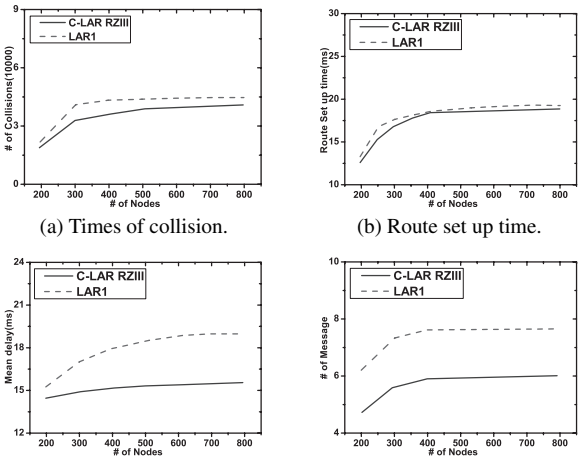
As a conclusion, there are two advantages using this scheme: (1) the area of RZII in C-LAR is much smaller than that of LAR. It means that RZII confines the route search to a smaller space. It indicates that the number of nodes involved the routing process is smaller, thus the overhead of route is smaller than that of LAR1. Meanwhile, according to the simulation results, it is also useful to decrease the packet collision, the route set up time and the mean delay; (2) RZII in C-LAR restrains the route request message to flood in a narrower space. It means that the request message is sent to D from S as directly as it can. Apparently, it provides a higher chance for S to select a short routing path to D.

4.5.3 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 when the Scenario III occurs, whatever S and D are in a same cluster or adjacent clusters, as shown in Fig. 14, S and D can ignore the hierarchical layers temporarily, S sends the request message without being passed through CHs. 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.

Comparison with LAR1

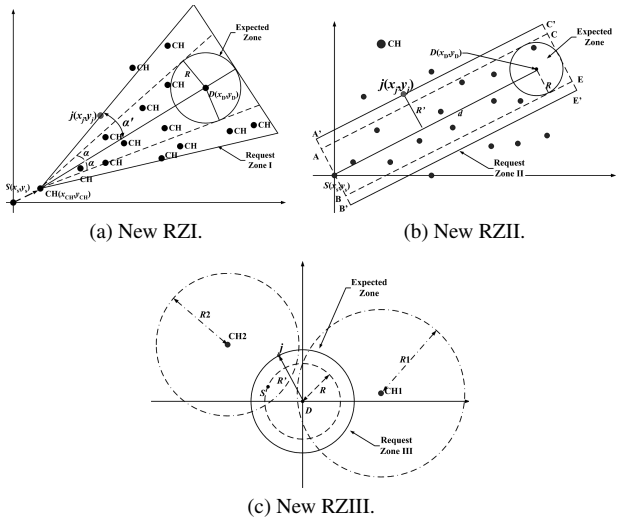
In this section, the same network configurations are used to compare the network layer performance of the C-LAR with LAR1, when S is within the expected zone. We observe from Fig. 15 that the mean delay and the normalized routing load in C-LAR are still lower than those in LAR1, which is caused by the route message flooding mechanism in C-LAR, resulting in more than one path exits between the node pairs. It leads to a more reliable route. Furthermore, the S and D can ignore the hierarchical layers temporarily, which also reduce the transmission delay which is caused by the S and D sending the request message and waiting for the reply message from the CHs. When using the LAR1, it is possible that too few relay nodes or no relay node exists because of the narrow route discovery space and node move-



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

(c) Mean delay. (d) Routing load.

Fig. 15 Network layer performance comparison in RZIII.



(a) New RZI. (b) New RZII.

(c) New RZIII.

Fig. 16 New request zone.

ment, which is easier to make a route broken down. When a broken link occurs in MANET, LAR1 tries search for new routes and re-establish the broken route. This behavior is detrimental to decreasing the mean delay and normalized routing load. On the other hand, we observe that route set up time and the total times of collision in both of two algorithms are much similar. The node pair (S, D) is in the same expected zone, thus the relative distance is quite short. It indicates the number of relay nodes in both algorithms is small, thus the route set up time in C-LAR is fairly close to that in LAR1. However, considering that the re-establish the broken route process increases a few packet collision times, so the total times of collision in LAR1 is still a bit more than that in C-LAR.

4.5.4 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 hole detection method is proposed.

In Scenario I, S forwards the request message to its CH, and then the CH sends it to any relay node i, i checks if there are next hop neighbor nodes locate within in the RZI by using the neighbor nodes' location information that are recorded in its NT. If there is no neighbor node suitable for being as the next hop, i returns the "Msg_Node_RErr" to the CH, which includes the location information of neighbor j which i considers is suitable for being as the next hop. After having received the message, CH will increase the angle α to α' and recalculate a new RZI. As shown in Fig. 16 (a), the line through CH(x_{CH}, y_{CH}) and $j(x_j, y_j)$ is:

$$Y = \frac{y_{CH} - y_j}{x_{CH} - x_j}\left(X - x_j\right) + y_j \tag{29}$$

Similarly, line through CH and D is given by:

$$Y = \frac{y_{CH} - y_D}{x_{CH} - x_D}\left(X - x_D\right) + y_D \tag{30}$$

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

$$\alpha' = \arctan\left|\frac{\left(\frac{y_{CH}-y_j}{x_{CH}-x_j}\right) - \left(\frac{y_{CH}-y_D}{x_{CH}-x_D}\right)}{1 + \left(\frac{y_{CH}-y_j}{x_{CH}-x_j}\right)\left(\frac{y_{CH}-y_D}{x_{CH}-x_D}\right)}\right| \tag{31}$$

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 having received the message Msg_CM_RErr, S will extend the line segment SA to SA', as shown in Fig. 16 (b). The new R' can be calculated as follows:

$$R' = \frac{\left|\left(x_D - x_S\right)x_j + \left(y_S - y_D\right)y_j + \left(x_S y_D - y_S x_D\right)\right|}{\sqrt{\left(x_D - x_S\right)^2 + \left(y_D - y_S\right)^2}} \tag{32}$$

In Scenario III, S will extend the radius of RZIII. The new radius r' is the distance from j to D, as shown in Fig. 16 (c).

4.6 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.

This procedure consists of seven steps:

Step 1. S calculates an expected zone by the approach we described in Sect. 4.2, while it uses the basic information of D which is exacted from the information <CH_ID(D), ID(D), GPS(x_D, y_D) >. Meanwhile, the estimated relative distance between S and D, $d(S, D)$, can be obtained.

Step 2. S judges the scenario:

```
1.  procedure Judge_scenario
2.  begin
3.    if(d(S, D) ≤ v_max[(t_1 − t_0) + Δt])
4.          then scenario III is initiated;
5.               return RZIII;
6.    else if(CH_ID(S) == CH_ID(D))
7.          then scenario II is initiated;
8.               return RZII;
9.          else scenario I is initiated;
10.              return RZI;
11. 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 <type, sour, dest, RZ(X), pathlist, h, routed>.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:

```
1.  procedure Establish_path
2.  begin
3.    Receive(Msg_Node_RReq);
4.    if(i ∉ RZ)
5.      then Discard(Msg_Node_RReq);
6.    exit(0);
7.    else if(i ∈ RZ)
8.      then case X of
9.          1: call Routing_RZI; break;
10.         2: call Routing_RZII; break;
11.         3: call Routing_RZIII; break;
12. end
13. procedure Routing_RZI
14. begin
15.   if(i == CH)
16.     if(D is i's CM)
17.     then add i to pathlist;h++;routed = 1;
18.     else
19.     then select next hop m from i's NT;
20.          and CH is preferred;
21.     call Judge(j);
22.   if(i == CM)
23.     if(D is in i's NT)
24.     then add i to pathlist;h++;routed = 1;
25.     else
26.     then select next hop n from i's NT;
27.          and CH is preferred;
28.     call Judge(j);
29. end
30. procedure Routing_RZII
31. begin
32.   if(CH_ID(i) == CH_ID(D))
33.     if(D is in i's NT)
34.     then add i to pathlist;h++;routed = 1;
35.     else
36.     then select next hop m from i's NT;
37.     call Judge(j);
38.   else
39.     then i drops the route request;
40. end
41. procedure Routing_RZIII
42. begin
43.   then select next hop j from i's NT;
44.   call Judge(j);
45. end
46. procedure Judge(j);
47. begin
48.   if(j ∉ RZ)
49.     then return Msg_Node_RErr to S;
50.   else if(j ∈ RZ)
51.     then send Msg_Node_RReq to j;
52.          add node i to pathlist; h++;
53. end
```

In the pseudo-code, we assume that the next hop is always closer to D. Until the "Msg_Route_RReq" reaches to D, routed = 1. In RZI, S and D are in different cluster, the main problem is that how to guarantee the strategy node i employed can pass through as many CHs, more precisely, virtual backbone nodes, as possible. When i chooses the next hop node, i prefers CH node in its NT (line 19, 20, 26, 27). In RZII, S and D are in same cluster. We firstly make sure that i is in a same cluster with D, because the rectangle type of request zone maybe cover some nodes which are not in this cluster (line 32, 38, 39). Then i checks its NT whether D is its adjacent node, and decides the next operation. In RZIII, request zone is equal to expected zone initially. The challenge is how to improve the efficiency. The strategy node i employed is that S and D can ignore the hierarchy temporarily (line 43).

Step 6.When "Msg_Route_RReq" has been received by D, it unicasts a route reply message along the reverse direction of the route that is 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 <CH_ID(D), ID(D), GPS(x_D, y_D), pathlist>.

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

4.7 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 [10], 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 a route failure has occurred. After having received the message, the source reinitiates a route discovery to search for a new routing path.

5. Performance Evaluation

In this section, we present simulation results to illustrate the performance of C-LAR protocol. The simulator is implemented within Global Mobile Simulation (GloMoSim) library by C++ language [29]. The GloMoSim library is a scalable simulation environment for mobile wireless network using parallel discrete-event simulation capability provided by PARSEC [30]. We tried to compare the performance of C-LAR with LAR scheme 1 (LAR1) that was implemented by J. Hsu and S.J. Lee and included within GloMoSim 2.03. The implementation of LAR1 followed the specification proposed in [10]. Other details are based on the discussions with Y.B. Ko. We examine two aspects of the C-LAR in simulations, namely, (1) the topology structure performance of clustering algorithm, which is as a basis of the C-LAR, (2) the comparison of the network layer performance of the well-known CBRP, VSR, LAR1, and TZRP with C-LAR.

In our simulation, all network nodes are located in a physical area of size $1000 \times 1000\,\mathrm{m}^2$ to simulate actual mobile ad hoc networks. The network size 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). For random waypoint, a node randomly selects a destination from the physical terrain, and then it moves in the direction of the destination in a speed uniformly chosen between the minimum and maximum roaming speed. After it reaches its destination, the node stays there for a specified pause time period. In our simulation, we conduct simulations for the RWP mobility models with a randomly distributed speed in the range from 5–30 m/s; the pause time is fixed to 30 seconds. The propagation path loss model used in our experiment 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 antenna height is hard-coded in the model (1.5 m). The radio bandwidth of each mobile node is 2 Mbps. Following [23], we assume that different frequency bands for the intra-cluster communication inside the individual clusters and the inter-cluster communication among adjacent clusterheads. Our simulation model considers the distributed coordination function (DCF) of 802.11, which employs carrier sense multiple access with collision avoidance (CSMA/CA) [26]. We do not employ request-to-send/clear-to-send (RTS/CTS) reservations for the RREQ packets to avoid the reservation overhead for these short packets.

The simulation time of each round lasts for 1000 seconds. Each simulation result is obtained from an average of the all simulation statistics. In each round, there are four application connections. The traffic generators used by the four application connections are constant bit rate (CBR). The CBR simulates a constant bit rate traffic generator. The generators initiates the first packet (i.e., start time) in different time and sent a 512 bytes packet each time.

5.1 Factorial Design

Factorial design is an experimental design technique especially useful to measure the effects of a group of factors on the output of an experiment. Applying this technique it is possible to determine that combination of the factor values which gives the best performance of the system. The complete analysis of the factors is called Full Factorial Design. Usually, a full factorial design is expensive, time consuming and not possible to carry out due to the huge number of combinations to be investigated. However, in most of the cases some of the factor values can be eliminated intuitively. In this paper, the cluster algorithm is based upon five parameters: node degree difference, average relative distance, average relative mobility, average link stability and residual energy. Based on the proceeding discussions, we propose the cluster algorithm which effectively combines all the five parameters with weighing factors w_1, w_2, w_3, w_4 and w_5, the value of which can be chosen according to the network applications. Note that the sum of these weighing factors is 1. For examples, in high speed scenario, average relative mobility and average link stability play very important roles in clustering the network, thus the weight of these factors should be made larger. And in field scenario where the battery power is the most important, the weight w_5 associated with average residual energy should be assigned largest.

We demonstrate the fractional factorial design with the help of the Table 1. The weight function is defined as an empirical mean value, where all these parameters are first normalized. All numeric values, as obtained from executing the cluster algorithm on a network consisted in 20 nodes, are tabulated. The values of these nodes' mobility are uniformly distributed in the range from 5–30 m/s. All nodes send and receive "Msg_Node_Hello" messages to/from their 1-hop neighbors. The node degree, which is defined as the total number of neighbors a node has, is shown in list 1. The node degree difference, Δ_i, of each node with ideal node degree $n^* = 4$ is computed in list 1. These nodes, which hear the broadcasted messages from its neighbors, get the RSSI of the two consecutive and successive messages. Once the neighbors list for all nodes are created, to these nodes, the average relative distance, the average relative speed and the average link stability can be obtained in list 2, 3 and 5, respectively. We choose some appropriate values for the residual energy of each node based on the stochastic model of a battery [31], [32]. A single battery cell is characterized by the open-circuit potential (V_{OC}), i.e., the initial potential of a fully charged cell under no-load conditions, and the cut-off potential (V_{cut}) at which the cell is considered discharged. For our experiments, we uses a Li-ion battery cell with the

Table 1 Execution of node weight.

Node i ID	Δ_i List 1	ξ_i List 2	θ_i List 3	ε_i List 4	μ_i List 5	W_i List 6
1	1	3	2	2	1	1.75
2	1	1	3	2	2	1.6
3	1	2	4	3	1	2.1
4	1	1	3	4	2	1.9
5	1	2	1	1	6	1.45
6	2	3	6	2	4	3.1
7	3	2	5	0	4	2.8
8	1	1	4	3	2	1.95
9	2	1	3	1	1	1.8
10	2	3	1	2	0	1.9
11	2	1	6	3	3	2.8
12	1	1	2	3	2	1.25
13	2	2	7	2	6	3.2
14	3	1	2	2	3	2.25
15	2	0	5	0	2	1.9
16	2	1	3	3	3	2.2
17	1	3	1	3	1	1.7
18	3	2	3	2	4	3.1
19	1	1	7	1	0	2.15
20	1	2	3	4	6	2.3

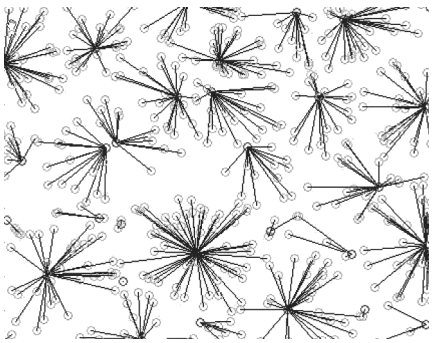


Fig. 17 Local output of simulation result.

following parameters: $V_{OC} = 4.3\,\text{V}$, $V_{cut} = 2.8\,\text{V}$, and each mobile device has 6 battery cells. The average residual energy, hereafter, can be calculated, which corresponds to list 4. After the values of all the components are identified, we compute the weighted metric, W_i, for every node as proposed in list 6 in our algorithm. The weighing factors considered are $w_1 = 0.4$, $w_2 = 0.2$, $w_3 = 0.2$, $w_4 = 0.15$ and $w_5 = 0.05$ The contribution of the individual components can be tuned by choosing the appropriate combination of the weighing factors.

5.2 Topology Structure Performance of C-LAR

Figure 17 shows that the local output of one of the simulation results of our algorithm at some time t. As shown in Fig. 17, the distribution of CHs is relatively uniform and the size of cluster is even. As mentioned earlier, the size of cluster needs to be small enough, but because of the dynamic movements of the mobile nodes, the perfect scenario can not be obtained easily.

Table 2 reports the performance results according to five usual clustering metrics:

Table 2 Clustering performance for different N.

N	$n - n^*$	CS	OH	OM	OHR	OMR
200	0.92	17.784	2.008	1.128	5.941	3.107
300	0.91	17.985	2.214	1.143	6.111	3.133
400	0.88	18.237	2.239	1.241	6.125	3.199
500	0.82	18.194	2.299	1.247	6.231	3.245
600	0.74	18.983	2.318	1.329	6.236	3.248
700	0.66	19.512	2.427	1.355	6.248	3.246
800	0.58	19.846	2.454	1.453	6.254	3.241

(1) average cluster size (CS) is the average number of CMs included in each cluster;

(2) average communication overhead of a CH per cluster formation (OH) is the average number of messages sent by each CH in cluster formation;

(3) average communication overhead of a CM per cluster formation (OM) is the average number of messages sent by each CM in cluster formation;

(4) average communication overhead of a CH per round (OHR) is the average number of messages sent by each CH in entire clustering process per round;

(5) average communication overhead of a CM per round (OMR) is the average number of messages sent by each CM in entire clustering process per round.

These metrics permit a more accurate evaluation of the quality of the obtained clustering structure are briefly observed. Table 2 shows that the CH number difference between n and n^* decreases as the total number of nodes increases. CS gauges the load imposed on CHs. The value of the metric increases slowly, which indicates that the cluster head election and cluster formation algorithm is effective and suitable for large scale MANET. OH and OM measure the overhead of communications between CMs and CHs in cluster formation, OHR and OMR indicate the overhead of communication between CMs and CHs in entire clustering process per round. The values of four parameters are stable, OH is stable at around 2.2, OM is at around 1.3, OHR is at around 6.1, and OMR is at around 3.2. That means C-LAR is efficient in controlling the communication overhead, and guarantees that various overheads do not increase acutely. The results demonstrate that our clustering algorithm performs well and is well adapted to meet its stated objectives in the environments for which it has been designed to operate.

5.3 Comparison with Four Protocols

C-LAR is compared with some famous and classical protocols, such as CBRP [13], VSR [14], LAR1 [10] and TZRP [9]. As discussed in Sect. 4.2, the expected zone has been re-defined as a correction for LAR1's definition by using a time interval Δt. In these simulations, C-LAR uses the revised definition of the expected zone; however, LAR1 still keeps the original.

Four performance metrics are introduced to evaluate the routing performance of C-LAR:

(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 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 18 shows the results of average end-to-end delay. From Fig. 18, CBRP shows 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 limits the nodes involved in the route discovery, and reduces the control packets. C-LAR 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 19 shows the results of success delivery ratio for C-LAR, CBRP, VSR, LAR1 and TZRP. It illustrates that C-LAR outperforms at any mobility speed, especially exhibited higher performance at higher speed. From Fig. 17, when the mobility speed = 30 m/s, because C-LAR has two methods to recovery 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 C-LAR may provide efficient fault tolerance in the sense of faster and efficient recovery from route failures in dynamic networks.

Figure 20 shows the results of discovery frequency performance. C-LAR 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 the source node's response. Thus, VSR also has more discovery times than that of C-LAR. 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 C-LAR 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.

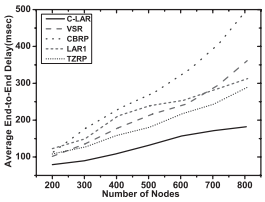


Fig. 18 Average end-to-end delay vs. number of nodes.

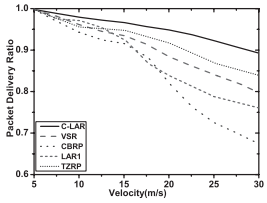


Fig. 19 Success delivery ratio vs. max.velocity.

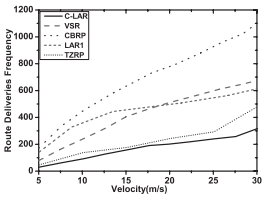


Fig. 20 Route discovery frequency vs. max.velocity.

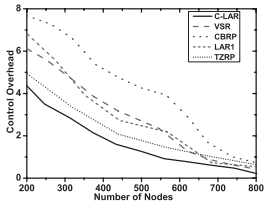


Fig. 21 Control overhead vs. number of nodes.

Figure 21 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. The total number of overheads per node among five protocols increase when the number of nodes increases. With a higher node density of MANET, the performance of each protocol remains unaffected. The simulation results show that the control overhead of C-LAR is lower than that of CBRP, VSR, LAR1 and TZRP, especially when the number of nodes increase large enough. By comparison, we can notice from Fig. 21 that the larger the size of the network is, the lower the control overhead of C-LAR is relative to the other four protocols.

6. Conclusion

In this paper, we have developed and analyzed a novel Cluster Based Location-Aided Routing Protocol for the design and operation of the large scale wireless mobile ad hoc networks. MANET is dynamic in nature due to the mobility of nodes. The association and dissociation of nodes to and from clusters perturb the stability of the network topology, and hence a reconfiguration of the system is often unavoidable. However, it is vital to keep the topology stable as long as possible. These clusterheads, form a virtual backbone in the network, determine the network's topology and stability. A weight-based clustering algorithm is used by C-LAR to establish a cluster cover of the networks and reduce routing control overhead and improve the networks scalability. This clustering algorithm takes into consideration the node degree, mobility, relative distance, battery power and link stability of mobile nodes. Moreover, using the location information of mobile nodes to assist routing can confine the route searching space into a smaller estimated range. The mechanism we adopted is to use geographical location information provided by GPS to assist routing. The location information of destination node is used to predict a smaller

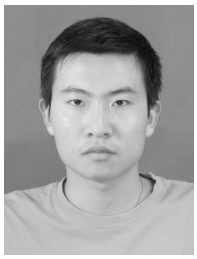
rectangle, isosceles triangle, or circle request zone, which is selected according to the relative location of the source and the destination, that covers the position of destination in the past. Instead of searching the route in the entire network blindly, C-LAR limits the search for a routing path to the so-called request zone, it is obvious that the smaller route discovery space reduces the traffic of route request and the probability of collision. Simulation results have shown that C-LAR outperforms other protocols significantly in route set up time, mean delay, routing overhead and collision, and simultaneously maintains low average end-to-end delay, high success delivery ratio, and low route discovery frequency.

References

[1] R. Wattenhofer, "Ad-hoc and sensor networks: Worst-case vs. average-case," International Zurich Seminar on Communications, pp.156–159, 2004,
[2] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks," Ad Hoc Networks, vol.2, no.1, pp.1–22, Jan. 2004.
[3] A. Boukerche, "Performance evaluation of routing protocols for ad hoc wireless networks," Mobile Netw. Appl., vol.9, no.4, pp.333–342, 2004.
[4] I. Chlamtac, M. Conti, and J.J.N. Liu, "Mobile ad hoc networking: Imperatives and challenges," Ad Hoc Networks, vol.1, no.1, pp.13–64, July 2003.
[5] D.B. Johnson, D.A. Malts, and Y.C. Hu, "DSR: The dynamic source routing protocol for multihop wireless ad hoc networks," IETF Internet Draft, 2004.
[6] E.M. Belding-Royer and C.E. Perkins, "Evolution and future directions of the ad hoc on-demand distance-vector routing protocol," Ad Hoc Networks, vol.1, no.1, pp.125–150, 2003.
[7] V.D. Park and M.S. Corson, "Temporally-ordered routing algorithm (TORA) version 1: Functional specification. Internet-Draft," draftietf-manet-tora-spec-00.txt, 1997.
[8] Z.J. Haas and M.R. Pearlman, "The performance of query control chemes for the zone routing protocol," ACM/IEEE Trans. Networking, vol.9, no.4, pp.427–438, 2001.
[9] L. Wang and S. Olariu, "A two-zone hybrid routing protocol for mobile ad hoc networks," IEEE Trans. Parallel Distrib. Syst., vol.15, no.12, pp.1105–1116, 2004.
[10] Y.B. Ko and N.H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom98), pp.66–75, 1998.
[11] B. Karp and H.T. Kung, "GPSR: Greedy perimeter stateless routing for wireless network," 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking, pp.243–254, 2000.
[12] S-C.M. Woo and S. Singh, "Scalable routing protocol for ad hoc networks," Wirel. Netw., vol.7, no.5, pp.512–529, 2001.
[13] M. Jiang, J. Li, and Y.C. Tay, "Cluster based routing protocol (cbrp)," draft-ietf-manet-cbrp-spec-01.txt, IETF, Internet draft version 01, 1999.
[14] F. Theoleyre and F. Valois, "Virtual structure routing in ad hoc networks," IEEE ICC 2005, Seoul, Korea, May 2005.
[15] L. Ritchie, H. Yang, A. Richa, and M. Reisslein, "Cluster overlay broadcast (COB): MANET routing with complexity polynomial in source-destination distance," IEEE Trans. Mobile Computing, vol.5, no.6, pp.653–666, 2006.
[16] J. Li, J. Jannotti, D.S.J. De Couto, D.R. Karger, and R. Morris, "A scalable location service for geographic ad hoc routing," ACM Mobicom 2000, 2000.
[17] G. Angione, P. Bellavista, A. Corradi, and E. Magistretti, "A k-hop clustering protocol for dense mobile ad hoc networks," ICDCSW'06, pp.10–15, 2006.
[18] M. Chatterjee, S.K. Das, and D. Turgut, "WCA: A weighted clustering algorithm for mobile ad hoc networks," Journal of Cluster Computing, vol.5, pp.193–204, April 2002.
[19] Y. Xu and W. Wang, "MEACA: Mobility and energy aware clustering algorithm for constructing stable MANETs," IEEE Milcom'06, 2006.
[20] P. Gupta and P.R. Kumar, "The capacity of wireless networks," IEEE Trans. Inf. Theory, vol.46, no.2, pp.388–404, 2000.
[21] G. Calinescu, I.I. Mandoiu, P.J. Wan, and A.Z. Zelikovsky, "Selecting forwarding neighbors in wireless ad hoc networks," Mobile Netw. Appl., vol.9, no.2, pp.101–111, 2004.
[22] J. Eriksson, M. Faloutsos, and S. Krishnamurthy, "Scalable ad hoc routing: The case for dynamic addressing," IEEE Infocom 2004, 2004.
[23] K. Xu and M. Gerla, "A heterogeneous routing procotol based on a new stable clustering scheme," IEEE Milcom 2002, pp.838–843, 2002.
[24] H. Ramin, Ad-hoc networks: Fundamental properties and network topologies, Springer, 2006.
[25] C.H. Edwards, Calculus with Analytic Geometry, Prentice Hall, Upper Saddle River, 1998.
[26] ANSI/IEEE std 802.11, 1999 Edition, Part 11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.
[27] P. Basu, N. Khan, and T. Little, "A mobility based metric for clustering in mobile ad hoc networks," IEEE ICDCSW 2001, pp.413–429, 2001.
[28] O. Dagedeviren and K. Erciyes, "A distributed backbone formation algorithm for mobile ad hoc networks," IEEE ISPA2006, LNCS 4330, pp.219–230, 2006.
[29] L. Gajaj, M. Takai, K. Tang, R. Bagrodia, and M. Gerla, "GlomoSim: A scalable network simulation enviroment," UCLA CSD Technical Report, #990027, 1999.
[30] R. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, B. Park, and H. Song, "Parsec: A parallel simulation environment for complex systems," Computer, vol.31, no.10, pp.77–85, 1998.
[31] P. Rong and M. Pedram, "An analytical model for predicting the remaining battery capacity of lithium-ion batteries," DATE 2003, IEEE Computer Society, 2003.
[32] T.D. Panigrahi, D. Panigrahi, C. Chiasserini, S. Dey, R. Rao, A. Raghunathan, and K. Lahiri, "Battery life estimation of mobile embedded systems," Fourteenth International Conference on VLSI Design, 2001.
[33] Z. Li, L. Sun, and E.C. Ifeachor, "Range-based relative velocity estimations for networked mobile devices," IEEE Trans. Veh. Technol., vol.58, no.3, pp.1–5, 2009.

Yi Wang received the B.S. degree in Department of Computer Science and technology from Tianjin University of Technology, Tianjin, P.R. China, in June 2003, and the M.S. degree in School of Electronics and Information Engineering from Xi'an Jiaotong University, Xi'an, P.R. China in June 2006. His research interests include wireless networks, mobile computing.

Liang Dong received the B.S degree in electronic engineering from Beijing University of Aeronautics and Asronautics, China, in 1997, the M.S degree in circuit and system from the Second Academy of China Aerospace, China, in 2000. He is now a Ph.D candidate in the Dep. of Elec. & Comp. Engineering, National University of Sinapore. He is with Healthcare department, Philips Research Asia-Shanghai, China. His research interests include wireless networks, image processing and video processing.



Taotao Liang received the B.S. degree in Department of Computer Science and technology from Zhengzhou University, Zhengzhou, P.R. China, now he is studying in School of Electronics and Information Engineering in Xi'an Jiaotong University, Xi'an, P.R. China, and will receive his mater degree in July 2009. His research interests include wireless networks, web services and distributed computing.

Xinyu Yang received his Ph.D. degree in School of Electronics and Information Engineering and was in postdoctoral position in Control Science & Engineering Station, Xi'an Jiaotong University, P.R. China. He is currently an associate Professor of Computer Science. His research interests include wireless networks, multimedia applications and multimedia network protocols.

Deyun Zhang received the BE degree in Computer Science from Xi'an Jiaotong University, Xi'an, P.R. China, in July 1964. He academically visited Osaka University, Osaka, Japan, from 1983 to 1985. He is currently a Professor of Computer Science, Xi'an Jiaotong University. His research interests include wireless networks, IPV6 network and distributed computing.