

# TOPOLOGY DISCOVERY IN MULTI-HOP CLUSTERING VEHICULAR NETWORKS

Ahmed Karam and Liren Zhang

College of Info. Technology, United Arab Emirates University  
Al Ain, United Arab Emirates

## **ABSTRACT**

*In this paper, we propose a novel multi-hop cluster architecture for location service protocol in vehicular ad hoc networks. The proposed scheme uses two parameters which are the connectivity between vehicles and the vehicle mobility to select cluster head. The performance of the proposed scheme is the tradeoff between the vehicle locations and the communication overheads. The proposed scheme is not only scalable but also reliable and is able to achieve high load balancing with fast convergence. The cluster constructed by the proposed scheme is more stable than existing vehicular ad hoc network clustering schemes. Specifically, the proposed scheme can increase the cluster-head lifetime up to 50%. The reason behind this achievement is that the proposed scheme considers vehicle mobility in terms of average link expiration time while selecting cluster-head.*

## **KEYWORDS**

*vehicle Ad hoc networks (VANETS), cluster, location services, communication, overhead*

## **1. INTRODUCTION**

In vehicular ad hoc network (VANET), the location information service can be obtained using either proactive location service protocols or reactive location service protocols, [1] [7] [11]. Typical location dissemination system approach includes the Distance Routing Effect Algorithm [7] for Mobility (DREAM), the GPS/Ant-Like routing algorithm (GPSAL) [8], the DREAM Location Service (DLS) [8] and the Simple Location Service (SLS). In this approach, hosts periodically exchange their location information, so that every host must maintain a location map of the whole network. The advantage of such system is robust and low cost for location query, since every host is acting as a location server. However, the disadvantage of these approaches is the large communication overheads for location updates. This makes the above approaches impractical in large scale network. Using reactive location service protocol [8] [12], host location information is obtained either from intermediate hosts or from the desired hosts on request basis. Clearly, this approach has low cost of exchanging location information, but higher cost for location searching. On the other hand, the network stability is usually weak since there is neither location map maintained in each host nor specific location servers available.

This paper focuses on hierarchical cluster-based location services protocol to provide accurate location information to mobile host in ad hoc network with good scalability. It is different from

some typical location service protocols being used in fixed geographic mobile networks such as grid [9] or circle [11]. Generally, cluster-based location services can be managed by either single-level or multi-level. In multi-level clustering, such as the Hierarchical State Routing (HSR) [19], maintenance of location map requires large communication overheads due to frequent and random change in multi-level hierarchical topology. On the other hand, packets forwarding consumes a large amount of bandwidth and requires high topology stability on the high-level backbone. Hence, multi-level cluster is unsuitable to ad hoc environment. In the contrary, the single-level clustering schemes are simple, since the cluster-head in these schemes only keeps track of local topology changes caused by host mobility. Therefore, the size of a cluster can be increased by  $k$ -hop cluster. This certainly can achieve better scalability than single-hop cluster. Inspired by the distance effect [3] and the virtual backbone quorum scheme [6], we propose a novel  $k$ -hop cluster-based location service (KCLS) protocol. The details of the propose approach is given in the next section.

## 2. THE KCLS PROTOCOL

A two-level logical hierarchy including intra-cluster level and inter-cluster level is used the KCLS protocol. The intra-cluster level is formed by cluster-head, which is acting as distributed location server to maintain an inter-cluster connection table using cluster IDs. Whereas, the inter-cluster level is formed by the members located within the same cluster. The members of inter cluster only hold information about local topology. Intra-cluster location management is based on a link state routing approach [13] within a cluster.

If any link of intra-cluster is either established or broken down, a Local Link Change (LLC) packet will be sent to all cluster members to update their LC table along with the link state changes. Based on the existing LC table, each cluster member builds an IntraR table based on shortest path Dijkstra's algorithm to indicate the next hop to reach each particular cluster member or neighboring cluster. The intra-cluster location management is robust and efficient at the same time because the routing table construction and link state exchange both are performed within the cluster. By contrast, the management of inter-cluster location information is controlled by cluster-heads. Each cluster-head maintains three tables: LC table, IntraR table and Location Service (LS) table. LS table lists the membership and inter-cluster connectivity. Furthermore, each row of LS table represents the Cluster State (CS) which includes the cluster ID, the list of cluster members, the sequence number of the location information, the neighboring clusters and the cluster-head's planar coordinates.

A cluster-head responds to location queries based on its LS table. In order to keep the accuracy of LS table in dynamic ad hoc networking environment, the location update rate must be fast enough to reflect topology changes. However, if forwarding of location update packet is based on flooding algorithm, then it produce large amount of overheads in such limited bandwidth environment. Thus, we developed an effective location update mechanism and integrated it in the proposed KCLS protocol. In the proposed location update mechanism, when a new cluster is created, a Cluster State (CS) packet will be sent by the cluster-head to all the other cluster-heads in the VANET. The CS packet carries the location information of the source cluster-head along with the cluster ID, cluster member list, neighboring cluster list and cluster-head's coordinates. Upon the reception of the CS packet, each cluster-head establishes its LS table or updates the location information of the cluster. On the other hand, the ad hoc network may cover a large area, the

sequence number of CS packets is used to eliminate duplicate copies and avoid delivery loops.

Since cluster-head is the local coordinator, which maintains and monitors its cluster topology. In this case, a cluster-head  $h_m$  may generate and transmit at most one CS packet based on time interval  $t_{CSm}$ , if and only if one of the following events is occurred during each  $t_{CSm}$ .

- Any member leaves or joins the cluster.
- A new neighboring cluster is connected, or an old neighboring cluster is disconnected.
- The cluster topology is changed due to new cluster creation, cluster-head election or cluster removal.
- Since the moment when the last CS packet was generated by  $h_m$ , the accumulated moving distance of  $h_m$  exceeds the predefined threshold  $D_{th}$

To reduce the overheads associated with transmitting CS packets with the whole image of configuration, the CS packets in the proposed KCLS protocol only contains the update information rather than the whole image of configurations. On the other hand, the time interval  $t_{CSm}$  for cluster-head  $h_m$  sending update CS packet is based on the cluster mobility pattern such as the average link available time. Hence, the cluster-head  $h_m$  can simply set the transmission interval using:

$$t_{CSm} = sT_{Am},$$

where  $s$  is the scaling factor, and  $T_{Am}$  is the average link available time of cluster  $C_m$ . According to the changes of LC table, the cluster-head can obtain  $T_{Am}$  by calculating the average link available time during a fixed time interval  $T_{SLOT}$ .

Note that although the cluster-head of sends out one CS packet in each interval and its neighboring clusters also receive the CS packet from in each time interval, but the other clusters, which are the neighboring clusters of, receive at most one CS packet of in every two-interval. This is because that after receiving two consecutive CS packets from, the neighboring clusters of only forward one CS packet to their neighboring clusters for updating the location information of. This is because the requirement for accurate location information between clusters decreases as their distance increases due to the distance effect [3]. The forwarding of CS packets is based on multicast mechanism, which requires the selected gateways to be able to connect multiple neighboring clusters simultaneously. This is a well-known set-cover problem [16], which can be done using a greedy algorithm based on the LC table in the cluster-head. That is, the host connecting to more neighboring clusters, excluding the neighboring cluster from which the CS packet is received, has the higher priority to be selected as gateway. This process continues until all neighboring clusters are covered. Then, the multicast of CS packets is through the selected gateways to the neighboring clusters.

The ad hoc network is divided on cluster basis using cluster ID, which is also the ID of cluster-head. Therefore, the entire network has a distributed location service system based on clustering architecture, in which each cluster-head takes the responsibility as location server to provide location information service to its cluster members. In this case, the location enquiry from an orphan host is only forwarded to the nearest reachable cluster-head. As a result, the location searching latency of KCLS protocol is much shorter than existing location service protocols

[6][14], in which clustering protocols search the destination across several databases. Given that some cluster-heads will be unreachable due to inter-cluster link failures, the distributed location service is still considered reliable. However, unreachable cluster-heads will only affect the relevant clusters whereas the remaining clusters in the VANET are still operational. In this case, data packet transmission is still possible even when the destination location information is not available. This is because that the location information in the packet head can be modified by intermediate cluster-heads along the path to the destination. For example, when an inter-cluster link of a data packet breaks before reaching the destination, then this data packet will be forwarded to the nearest active cluster. The cluster-head then modifies the location information, which includes destination cluster ID, the sequence number and the coordinates of cluster-head. Then, the route of the packet to the destination is modified accordingly. Furthermore, the KCLS protocol is capable of improving the performance of routing protocols by increasing route stability, enhancing route recovery capability and reducing overheads. For example, according to proactive routing protocols such as Dynamic Source Routing (DSR) [17], packets must carry a full routing list from source to destination, which generates a large amount of overheads, especially for long routing paths spanning over large scale network. On the contrary, in KCLS protocol, cluster-head decides a cluster level route to the destination using the LS table using Shortest Path algorithm. Since the response for a location enquiry is sent back to the source host by piggybacking, then data packet only needs to carry a cluster level route, which has much less overheads comparing to host level route. Also, cluster level route is more stable and adaptable for dynamic ad hoc topology comparing to host level route. In this case, the breakage of intra-cluster links has slight effect on forwarding packet, since several alternative paths to the next cluster can always be found. If the expected cluster on a route is unreachable, instead, the data packet is forwarded to the nearest available cluster for modifying the location information and searching for possible cluster level route. Therefore, the KCLS protocol is able to self-discover cluster level route against link failure. Accordingly, long latency due to route recovery process is avoided.

### 3. PERFORMANCE ANALYSIS

Simulations are conducted to evaluate the performance of KCLS protocol. The ad hoc network used in the simulation has homogeneous mobile hosts, which are randomly distributed in the area with square units. The transmission range of all nodes in the network is fixed and it is equal to unit. The connection arrival rate for each host is assumed to be Poisson distributed with a mean value of  $\lambda$ . The famous random walk mobility model is used to simulate the node mobility in the simulation. In random way model, the host speed is changed only at the beginning of each host-moving epoch. The moving rate and direction of a host within an epoch are constant, following uniformly distribution among predefined ranges  $v_{min}$  and  $v_{max}$  respectively, where  $v_{max}$  is the maximum rate of host moving. All clusters are created and maintained by KCMBC approach. The simulation parameters used in the paper are shown in Table 1. The simulation is independently repeated 10 times. In each simulation run, 50,000 random connection requests are generated excluding the warm-up period of 1,000 random connection requests. The warm-up period is used to ensure that the results are estimated on the basis of a steady simulation process. To ensure the accuracy of simulation results, the confidence interval is calculated as that at least 95% of estimated values for  $\lambda$  obtained from these 10 simulation runs fall in the interval  $(\lambda - \delta, \lambda + \delta)$ .

Table 1. Simulation Parameters for Location Service Protocol

Items	Value
Total number of hosts $N$	1000~4000
Maximum moving rate $V_m$	0.2~1 unit/s
Time interval of host moving epoch $t_e$	1s
Mean connection arrival rate $l_{Call}$	1/420s
Scaling factor $s$	0.1~0.5
Time interval for measuring link available time $T_{SLOT}$	10s
Distance threshold $D_{th}$	2 units

Table 2 shows the average number of clusters created in the ad hoc networks with different simulation configurations. As can be seen from Table 2 that for any fixed value of  $N$ , the average number of clusters decreases as the value of  $k$  increases, especially in large host density networks.

Table 2. The average numbers of clusters in the ad hoc network

$N$ $K$ Hops	1000	2000	3000	4000
1	377.4	589.9	617.3	722.0
2	257.1	272.1	260.0	251.8
3	206.2	159.4	151.4	152.3
4	170.1	115.0	91.2	90.7

#### 4. OVERHEADS IN THE INITIAL STAGE

In cluster initial stage, each cluster-head broadcasts a CS packet to the other cluster-heads when clusters are constructed. Let  $d_c$  denote the average number of neighboring clusters around one cluster,  $N_h$  denote the average number of hosts in a cluster,  $N_c$  denote the total number of clusters in the network, and  $H_c$  be the average number of hops for an arbitrary intra-cluster route. Upon receiving a CS packet, the cluster-head multicasts the CS packet to other neighboring clusters. Since each CS packet is forwarded by every cluster for one time, then each CS packet is transmitted by hosts totally  $d_c H_c N_c$  times. If  $H_c$  is approximated as  $k$ , and  $N_c = N / N_h$ , the overheads caused by CS packets in the initial stage are given by:

$$O_{CS} = d_c H_c N_c^2 \gg d_c k N^2 / N_h^2$$

Moreover, the overheads generated in the initial stage also include the overheads for cluster formation. Recall Section II.A, the total overhead for cluster formation is  $O_{CF} = (2k + 3)N$  packets. Therefore, the total overheads created in the initial stage can be calculated as follows

$$O_{Initial\_KCLS} = O_{CF} + O_{CS} \gg (2k + 3)N + d_c k N^2 / N_h^2$$

As can be seen from Fig 1, the overheads significantly decrease as the value of  $k$  increases. This is due to the fact that large value of  $k$  results in large cluster size and less number of clusters in the

network. However, for a fixed value of  $k$ ,  $O_{Initial\_KCLS}$  increases as the number of hosts  $N$  increases, but a large  $k$  can suppress the increase rate of  $O_{Initial\_KCLS}$  while  $N$  increases.

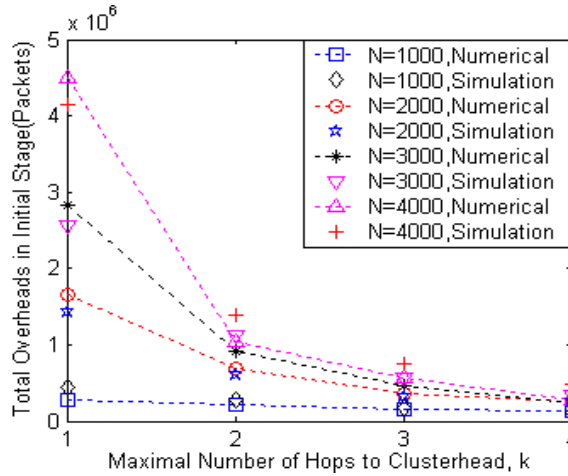


Fig 1. The overheads created by KCLS in the initial stage

## 5. COST IN LOCATION MAINTENANCE STAGE

There are three main overhead in location maintenance stage: intra-cluster location updates overhead, inter-cluster location updates overhead and periodical beacons overhead. From the performance comparison point of view, since beacons have been used by many routing protocols in MANETS, the overheads of beacons is not necessary and therefore will not to be discussed here. The overheads for intra-cluster location updates are caused by intra-cluster link activation and deactivation. Any change in intra-cluster links will result in generating a LLC packet that will be sent to each cluster member. On the other hand, such intra-link changes may affect the following: cluster membership, cluster removal, cluster re-election, cluster structure such as cluster merge, and the inter-cluster connectivity. Therefore, the CS packets are sent out to update location information. Beside of this, some small amounts of overheads are created by location enquiries. Therefore, the cost of location management, in terms of control packets per second, is a summation of the cost of inter-cluster location update ( $C_{inter}$ ), the cost of intra-cluster location update ( $C_{intra}$ ), and the cost of location enquiry ( $C_{enq}$ ). The total cost of location management is given by:

$$C_{KCLS} = C_{intra} + C_{inter} + C_{enq}$$

in which the small cost  $C_{enq}$  can be neglected, since every location enquiry packet is only transmitted within one cluster.

Fig. 2 shows the total cost of location management  $C_{KCLS}$  versus  $k$  when  $V_m = 0.5$  unit/sec and  $s = 0.2$ . It is obvious that the total cost increases with the number of hosts  $N$ . However, a large value of  $k$  can suppress the increase rate of  $C_{KCLS}$  while  $N$  increases. For example, when  $k = 1$ , the total cost for  $N = 4000$  is about 22.3 times of that for  $N = 1000$ ; but when  $k = 3$ , the total cost for  $N = 4000$  is only 11.6 times of that for  $N = 1000$ . It can be found that when the value of

$k$  is small, the total cost is mainly determined by  $C_{inter}$ . With the increase of  $k$ ,  $C_{inter}$  decreases, whereas  $C_{intra}$  increases and performs more and more contribution to the total cost. However, when  $N$  has a fixed value, the total cost decreases with the increase of  $k$ . It also can be seen that the cluster structure with  $k = 1$  is not scalable for a large diameter ad hoc network. As shown in Fig. 2, when  $k$  is ranging from 2 to 4, the total cost of location management can be significantly reduced.

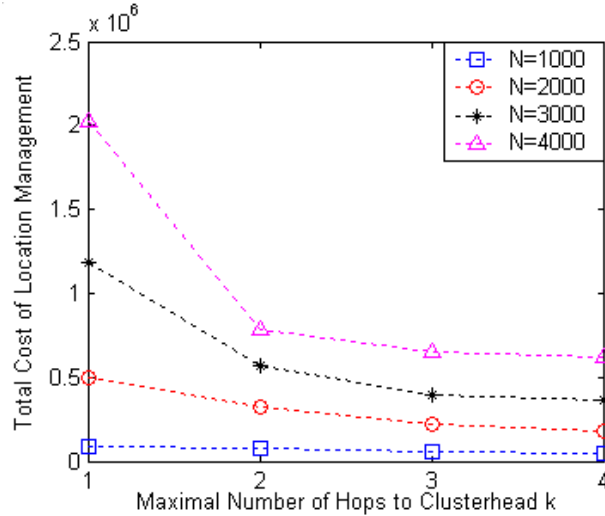


Fig 2. The Total Cost of Location Management,  $V_m = 0.5$  Unit/Sec &  $s = 0.2$

Fig. 3 shows the effect of host mobility on  $C_{KCLS}$ . It is clear that  $C_{KCLS}$  increases with  $V_m$ , since both link states and cluster structures change frequently while the rate of host moving becomes large. However, a large value of  $k$  is able to suppress the increase of  $C_{KCLS}$  while the degree of host mobility increases. From Fig 5, we can find that cluster structures with a large  $k$  are able to significantly reduce the cost of location management, especially when host moving is drastic.

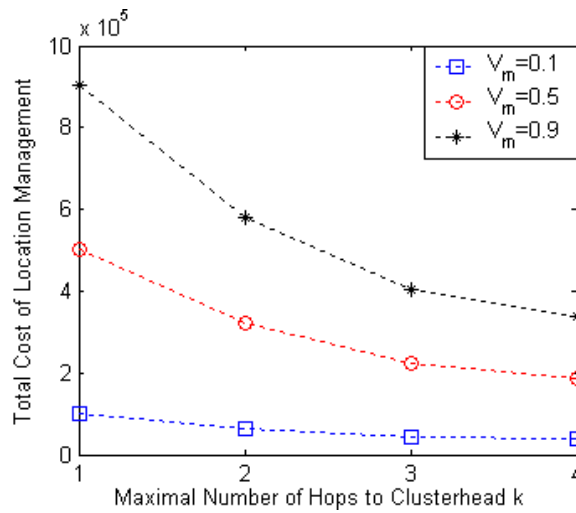


Fig 3 The effect of host mobility on  $C_{KCLS}$ ,  $s = 0.2$  and  $N = 2000$

Table 3. Cost Comparison Between Lsr And Kcls Unit/Sec, And.

$N$	1000	2000	3000	4000
$C_{LSR}$	$8.04 \cdot 10^5$	$6.44 \cdot 10^6$	$2.17 \cdot 10^7$	$5.15 \cdot 10^7$
$C_{KCLS}$	$0.55 \cdot 10^5$	$2.21 \cdot 10^5$	$3.98 \cdot 10^5$	$6.50 \cdot 10^5$

The KCLS protocol can be considered as a hierarchical link state protocol, in which the global inter-cluster link states are maintained by cluster-heads while the local intra-cluster link state is stored by each cluster member. Thus, it is interesting to compare the cost of location management under KCLS with the cost of link state updates in the LSR protocol[4],[13], which are denoted as  $C_{KCLS}$  and  $C_{LSR}$ , respectively. When  $V_m = 0.5$  unit/sec,  $k = 3$  and  $s = 0.2$ , as shown in Table 6,  $C_{LSR}$  increases with  $N$  following an approximately square rate, which is expected since any link change triggers a location update packet that must be forwarded by every host once. From Table 3, it also can be found that  $C_{KCLS}$  is much less than  $C_{LSR}$  for a fixed  $N$ . For example, when  $N = 4000$ ,  $C_{LSR}$  is about 80 times of  $C_{KCLS}$ . The total cost of KCLS increases with  $N$  following a linear rate. The average cost for location management charged by each host is a sub-linear function of host density.

### 6. ACCURACY OF LOCATION SERVICE

The accuracy of location service can be evaluated by the average hit probability of the location information obtained by an enquiry. Specially, the hit probability is defined as the probability that the response of a location enquiry is able to provide the correct cluster ID of the destination, where denotes that cluster-hop distance exists between the source-destination pair.

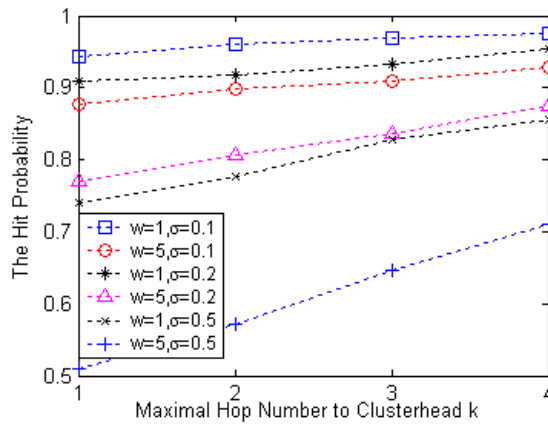


Fig 4. The Hit Probability Versus  $k$ ,  $N=2000$  &  $V_m=0.5$  Unit/Sec

Fig 4 illustrates the average hit probability versus  $k$  by using the KCLS protocol, when  $N = 2000$  and  $V_m = 0.5$  unit/sec. It can be seen that  $P_{HC}(w)$  increases with an approximately linear rate when  $k$  increases, since host sojourn time in a cluster increases while the average cluster size increases. Fig 4 also shows that the average hit probability decreases when the cluster-hop distance



$w$  changes from 1 to 5. This is because that the neighboring clusters of the destination's cluster  $C_m$  can receive at most one CS packet of  $C_m$  every  $t_{CSm}$  interval, but other clusters can only get at most one CS packet of  $C_m$  every  $2t_{CSm}$  interval. Thus, stale location information attached in data packets can be revised by the intermediate cluster-heads, which are close to the destination.

From Fig 4, it also can be found that when both  $w$  and  $k$  are fixed, the hit probability decreases with the increase of the scaling factor  $s$ . This is because that the latency for the transmission of CS packets is positive proportional to  $s$ . On the other hand, as discussed in Section IV.B, a small  $s$  results in frequent inter-cluster location updates, and causes a large amount of control overheads. It is obvious that tradeoff exists between the cost of location management and the accuracy of location service. As the neighboring clusters around the destination maintains the latest location information, the value of  $s$  can be determined by  $P_{HC}(w = 1)$ , the average hit probability of the response for a location enquiry that performs in the destination's neighboring clusters. To ensure  $P_{HC}(w = 1)$  higher than 90% for any  $k$ ,  $s$  is set as 0.2.

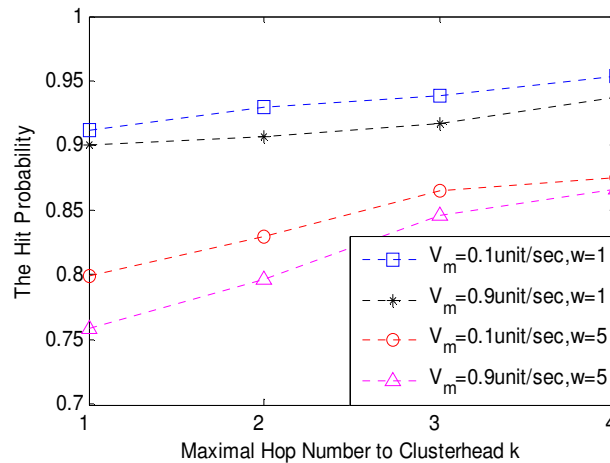


Fig 5. Impact of Host Mobility on The Hit Probability,  $s = 0.2$  &  $N = 2000$

Fig 5 shows the hit probability  $P_{HC}(w)$  when the maximal host moving rate is 0.1 unit/sec and 0.9 unit/sec, respectively. As it can be seen from Fig 5 that for both scenarios, the hit probability increases with the increment in the value of  $k$ . Furthermore, the figure also shows that the hit probability decreases as the value of  $w$  increases. Moreover, when  $w$  has a fixed value, there is a little difference between  $P_{HC}(w)$  for high host moving rate scenario ( $V_m = 0.9$  unit/sec) and  $P_{HC}(w)$  for low host moving rate scenario ( $V_m = 0.1$  unit/sec). It is clear that the change of maximal host moving rate makes little influence on the hit probability. This is because in the KCLS protocol, the transmission interval for CS packets is proportional to the average link available time of corresponding clusters, which can reflect the degree of host mobility. Thus, KCBL is able to adapt the frequency of location updates to various dynamic environments.

## 7. CONCLUSION

This paper presents a novel location service protocol based on clustering architecture. The proposed protocol has the following advantages over existing schemes. (1) It is able to

significantly reduce the overheads by the use of simple cluster level route using shortest distance algorithm combined with self-determined multicast inter-cluster forwarding based on cluster mobility pattern or group characteristics. (2) The clustering architecture based on cluster ID has flexible scalability in support of large scale ad hoc network. (3) It has the capability of cluster level self-route discovery against the inter-link failures. (4) Based on the distance effect, it is able to provide more accurate location information within the cluster and nearby neighborhoods, which matches the dynamic nature of ad hoc network very well.

The simulation results show that as the value of  $k$  increases the total cost in the location maintenance stage and the overheads in the initial stage decreases. A large value of  $k$  has the following advantages: first, it is able to suppress any increment in the total cost when the number of host in the network increases. Second, it is also able to increase the hit probability for location services. Third, it has the ability to reduce the passive effect of host mobility. Moreover, scaling factor  $\alpha$  is also used to balance the cost and the accuracy of location service. With optimal values of  $k$  and  $\alpha$ , the total cost of using KCLS is two percent less than the total cost of using link state protocol.

## REFERENCES

- [1] T. Camp, Location Information Services in Mobile Ad Hoc Networks, Handbook of Algorithms for Mobile and Wireless Networking and Computing, Chapter 14, pages 317-339, 2005.
- [2] I. Stojmenovic, Position-based routing in ad hoc network, IEEE Communic. Magazine, pp.128-134, July 2002
- [3] S. Basagni, I. Chlamtac, and V.R. Syrotiuk, Geographic Messaging in Wireless ad hoc Networks, Proc. 49th IEEE VTC, Vol.3, pp. 1957–1961, 1999
- [4] M. Joa-Ng and I.T. Lu, A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks, IEEE JSAC, Vol.17, No.8, pp.1415-1425, 1999
- [5] A. Amis, R. Prakash, T. Vuong, and D.T. Huynh. Max Min D-Cluster Formation in Wireless Ad hoc Networks, Proc. IEEE INFOCOM, Mar. 1999
- [6] Z.J. Haas and B. Liang, Virtual backbone generation and maintenance in ad hoc network mobility management, Proc. of IEEE INFOCOM 2000.
- [7] S. Basagni, I. Chlamtac, and V.R. Syrotiuk, B.A. Woodward, A distance routing effect algorithm for mobility (DREAM), Proc. MOBICOM, pp.76-84,1998
- [8] T. Camp, J. Boleng, and L. Wilcox, Location information services in mobile ad hoc networks. Proceedings of the IEEE International Conference on Communications (ICC), pp.3318-3324, 2001
- [9] J. Li and et al, A Scalable Location Service for Geographic ad hoc Routing, Proceedings of the 6th international conference on Mobile computing and networking, ACM, pp. 120 -130, 2000
- [10] KH Wang, and B. Li. Group Mobility and Partition Prediction on Wireless Ad-Hoc Networks. In Proc. of IEEE ICC Conference, pp.1017-1021, 2002
- [11] K.N. Amouris, and S. Papavassiliou, M. Li, A position based multi-zone routing protocol for wide area mobile ad-hoc networks, Proc. 49th IEEE VTC, pp.1365-1369, 1999
- [12] M. Kasemann and et al, A reactive location service for mobile ad hoc networks. Technical report, Department of Science, University of Mannheim, TR-02-014, November 2002
- [13] R. Perlman, Interconnections: Bridges and Routers. Reading, MA: Addison-Wesley, pp. 149–152, 205–233, 1992
- [14] L. Blazevic and et al, Self-organization in mobile ad hoc networks: the approach of terminodes, IEEE Comm. Magazine, pp.166-175, June 2001
- [15] S.-C. Woo and S. Singh, Scalable routing protocol for ad hoc networks. ACM Wireless Networks, Vol7, No5, pp.513--529, 2001
- [16] T. Cormen, C. Leiserson, and R. Rivest, Introduction to Algorithms, MIT Press, Cambridge MA, 1990

- [17] D. B. Johnson and D. A. Maltz, Dynamic Source Routing in ad hoc Wireless Networks, Mobile Computing, edited by Tomasz Imielinski and Hank Korth, Paper 5, Kluwer Academic, pp. 153-181, 1996
- [18] Supeng Leng, Liren Zhang and Yifan Chen, k-hop Compound Metric Based Clustering Scheme for Ad Hoc Networks, Proc. of IEEE ICC 2005, Korea
- [19] J. Sucec and I. Marsic, Hierarchical Routing Overhead in Mobile ad hoc Networks, IEEE Trans. on Mobile Computing, Vol.3, No.1, 2004
- [20] S. Capkun, M. Hamdi, and J.P. Hubaux, GPS-free positioning in mobile ad-hoc networks, Proc. Hawaii Int. Conf. on System Sciences, Jan. 2001.
- [21] S Leng and L Zhang, "A Novel k-Hop Compound Matric Based Clustering Scheme for Ad Hoc Wireless Networks", IEEE Trans on Wireless Communications, Vol. 8, No. 1 , 2009.
- [22] J. Tang, L. Zhang and S. Siew, "Providing Differentiated Services Over Shared Wireless Down Likes through Buffer Management", IEEE Trans on VT, Vol. 57, No. 1, 2008.
- [23] Y. Zhou, G.S Poo and L. Zhang, "Dynamic Multicast Routing and Wavelength Assignment Using Generic Graph Model", IEE Trans. on Communications, Vol. 12, No.7, 2008.