

Efficient Location Services Using Hierarchical Topology of Mobile Ad Hoc Networks

Prasad Naik Hamsavath, G. V. Singh

School of Computer and Systems Science, Jawaharlal Nehru University, New Delhi, India.
Email: naikphd@gmail.com, gvs10gvs@yahoo.co.in

Received September 21st, 2011; revised October 20th, 2011; accepted October 30th, 2011

ABSTRACT

The thesis work was carried out in two phases. Firstly, we have designed the Cluster-Based Object Location Services (CBOLS) for finding the object location in a cluster based network topology. Secondly, to take the advantages of clustering architecture of Mobile Ad hoc Networks, we have especially designed Object Location Clusters Algorithm (OLCA) for efficient Object Location management in MANET. Our comparison of simulation analysis shows that the CBOLS is more efficient and accurate in location information services and more robust than GLS and HRLS. In addition, we believe that our comparative study with the various location services schemes was facilitated accurate results in ad hoc networks. We are portioned our network area by designing a new clustering algorithm called OLCA (Object Location Clusters Algorithm) for efficient object location in Mobile Ad hoc Networks. Simulation results and comparative analysis shows the performance and effectiveness of CBOLS and OLC.

Keywords: Location; Clusters; Objects; Nodes; MANET

1. Introduction

Location information has recently been applied to MANET [1]. There are three types of Location Services available with the Mobile Ad hoc Networks. Such are Proactive, Reactive and Hybrid as explained at **Figure 1**.

In order to provide end-to-end communication throughout the network, mobile nodes must cooperate in handling network topology functions. It is very challenging issue in order to maintain the location information of the mobile hosts due to absence of centralized/dedicated servers in Mobile ad-hoc networks. Therefore location management becomes an important issue.

2. Literature Review

Routing protocols [2-6] are studied as a part of our research and an important subject. A variety of location based routing protocols [7-12] are exist and these protocols are of good scalability, and less overhead. These protocols usually assume that its part of the algorithm for obtaining location's information with the help of a global position system (GPS) [13]. Though these protocols seem to offer better location services, but they increase the routing delay time due to updation of location services as part of its algorithm. So, it is necessary to design an efficient algorithm for better location updates and searches. Clustering for efficient location services in MANET by

utilizing the benefits of clusters for achieving the better throughput and performance.

After study of literature review, we have exclusively proposed a cluster based location services protocol called Cluster Based Object Location Services (CBOLS) for efficient location updates and searches.

3. CBOL Algorithm

- 1) Initialize LOCN packet;¹
- 2) Initialize LACK packet;
- 3) Initialize LREQ packet;
- 4) Initialize LREP packet;
- 5) int Location Registration() {
- 6) broadcast LOCN to its neighbors;
- 7) Acknowledge LACK who receive LOCN;
- 8) for (LOCN=0; LOCN ≤ its neighbors; LOCN++) do;
- 9) if (LOCN received = true) do;
- 10) send LACK;
- 11) return LACK;
- 12) else
- 13) repeat step 8; }
- 14) call Location Query();
- 15) int Location Query(){
- 16) do

¹LOCN = Location; LACK = Location Acknowledgement; LREQ = Location Request; LREP = Location Reply; CH = Cluster Head; GW = Gateway.

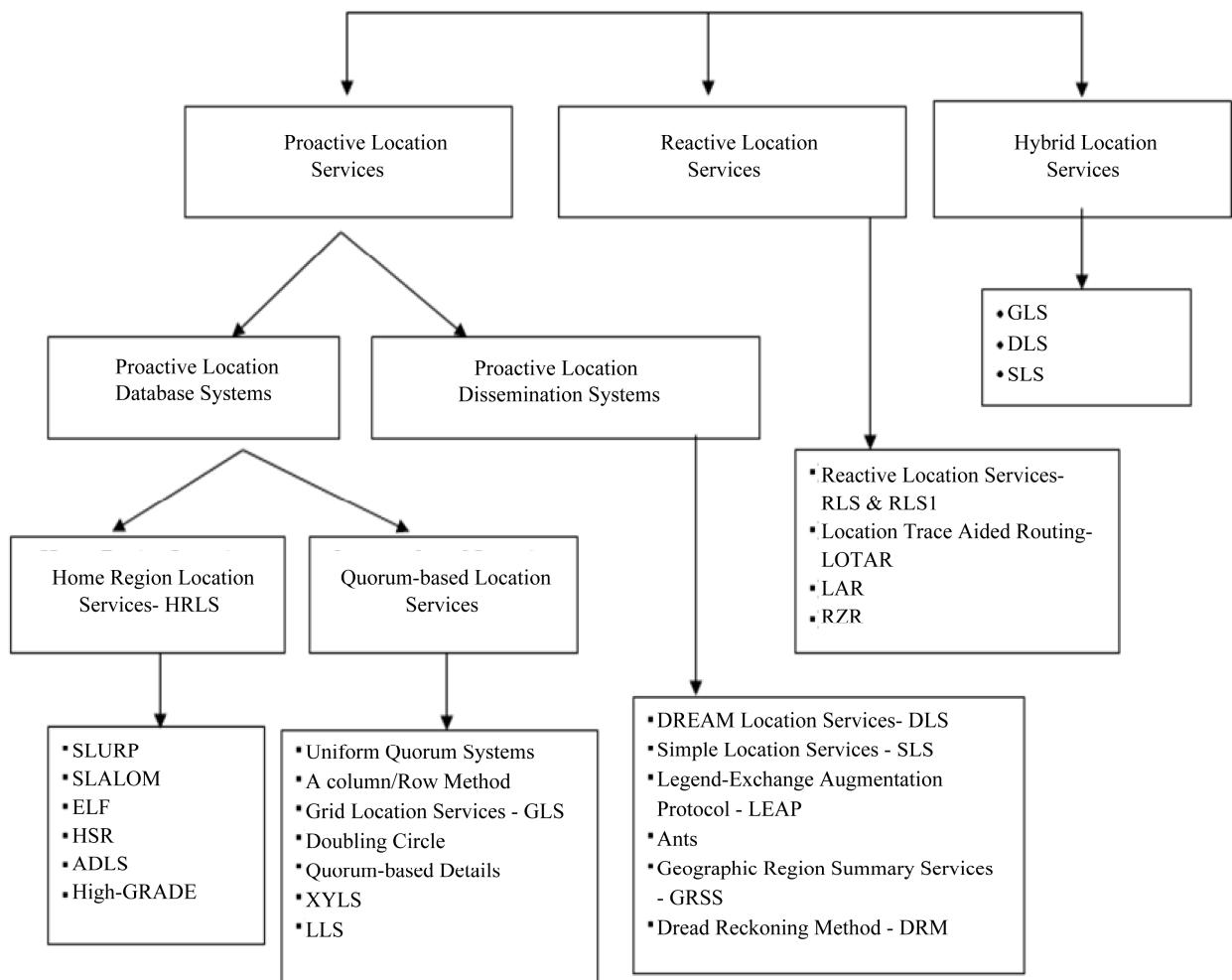


Figure 1. Types of location services in MANET.

- 17) broadcast LREQ to its destination node via its Clusterheads;
- 18) record its field in Location_table before sending LREQ;
- 19) if (CH received LREQ = true) do
- 20) record its LREQ field in its Location_table;
- 21) if (check LREQ field seen previously = true) do
- 22) discard LREQ packet;
- 23) else if (LREQ field & CH Location_table field = true) do {
- 24) send LREP packet;
- 25) return LREP;
- 26) Terminate(); }
- 27) else if (LREQ field & CH Location_table field != true) do
- 28) send LREQ packet to its nearest CH via Gateway;
- 29) check if (two Gate exists = true) do
- 30) Find the shortest path using CH routing table entries;
- 31) Send LREQ packet to its nearest Gateway having highest weight;
- 32) if (GW received LREQ = true) do
- 33) record its LREQ field in its Location_table;
- 34) if (check LREQ field seen previously = true) do
- 35) discard LREQ packet;
- 36) else if (LREQ field & GW Location_table field = true) do {
- 37) send LREP packet;
- 38) return LREP;
- 39) Terminate(); }
- 40) else if (LREQ field & GW Location_table field != true) do
- 41) send LREQ to its nearest CH;
- 42) if (CH received LREQ = true) do
- 43) record its LREQ field in Location_table;
- 44) if (check LREQ field seen previously = true) do
- 45) discard LREQ packet;
- 46) else if (LREQ field & CH Location_table field = true) do {
- 47) send LREP packet;
- 48) return LREP;
- 49) Terminate(); }
- 50) else

```

51) repeat step 23 - 49; }
52) void Terminate();{
53) Exit Application(); }
```

4. Simulation

For simulation purposes, we have used GloMoSim (Global Mobile Information System Simulator-Zeng *et al.*, [14] 1998). GloMoSim is a discrete event parallel environment for large wireless and wireline communication networks. GloMoSim uses a parallel discrete-event simulation capability provided by PARSEC (PARallel Simulation Environment for Complex systems) (Bagrodia [15], 1998). PARSEC is a C-based discrete-event simulation language developed by the Parallel computing laboratory at UCLA, for sequential and parallel execution of discrete-event simulation models. It can also be used as a parallel programming language. GloMoSim is developed at UCLA (California, USA) and is the second most popular wireless network simulator.

5. Performance Analysis of CBOLS

In this simulation work we have compared the metrics with Grid Location Services (GLS) and Home Region Location Services (HRLS) with our proposed Cluster-Based Object Location Services (CBOLS). Our simulation result shows the average location registration cost at 550 nodes at the mobility rate at 10 m/s compared with GLS and HRLS.

The below **Figure 2** depicts that our CBOLS (Cluster-Based Object Location Services) shows the minimum registration cost when compared with GLS and HRLS. Even though the GLS and HRLS results are similar, the CBOLS show better result as the cost of a node to register in the network is very low. Each node in the network gets registered with each cluster while the cluster is being formed.

Figure 3 shows the average location update cost of all the nodes in the network. In CBOLS, we see, the location update cost grows much more slowly than GLS and HRLS so we can define CBOLS update cost as $O(v \log N)$ vs $O(v \sqrt{N})$. When the number of nodes (N) of the network density (order O) grows, the number of location update cost ($v \log N$) also grows along with the varied network density.

Figure 4 shows that the average location finding cost at each node differs with High Region Location Service (HRLS) which is basically without cluster based network. This is a usually a static network where all the nodes are stable in routing and location information services. Whereas, the GLS and CBOLS illustrate similar results with location finding cost at each node, CBOLS exhibits better result than GLS, in its overall finding cost scale.

The **Figure 5** shows that the average location mainte-

nance cost at each cluster is minimum with CBOLS, compared to GLS and CBOLS show similar results. The location maintenance cost grows as the number of nodes in the network increases. Moreover HRLS cost is very high and performs very poor as node density grows.

The **Figure 6** illustrates the average control overhead

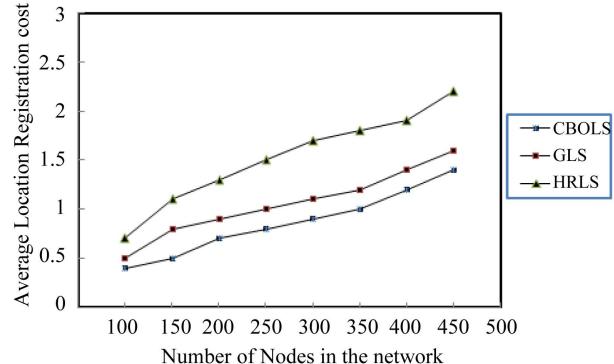


Figure 2. Average location registration cost per node in the network.

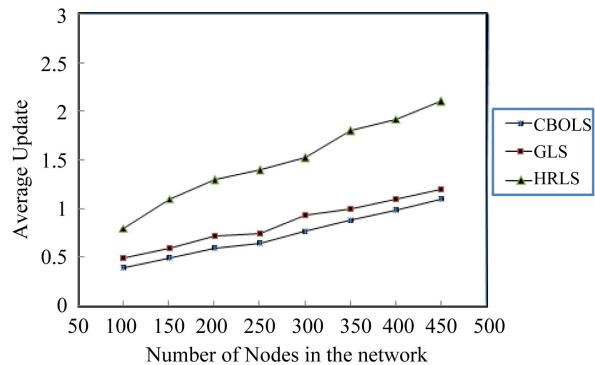


Figure 3. Average location update per each node in the network.

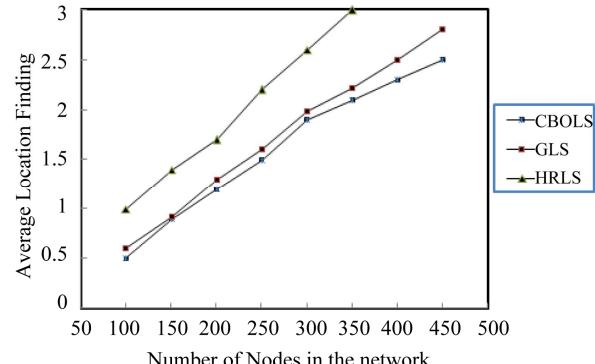


Figure 4. Average location finding cost at the mobility of the node up to 10 m/s (about 22 miles per hour). Each line corresponds to a different movement threshold.

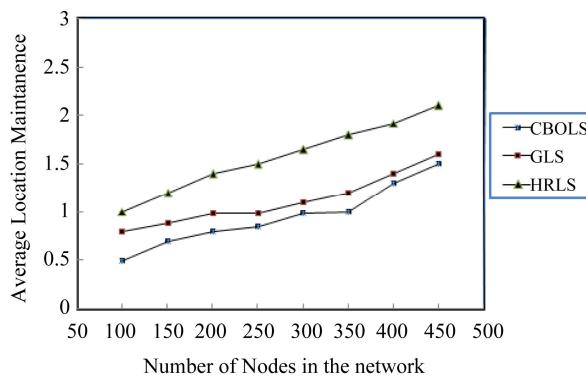


Figure 5. Average location maintenance cost at the mobility speed of 10 m/s.

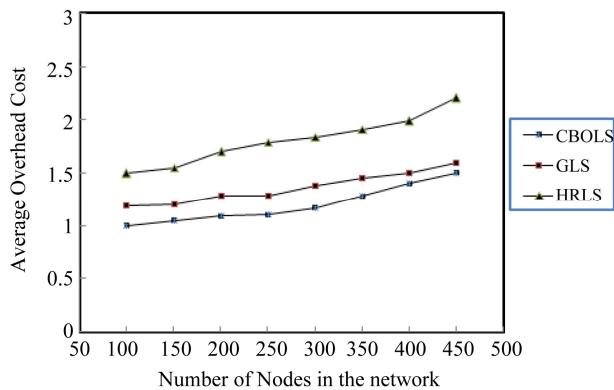


Figure 6. Average overhead cost at 500 nodes in the network.

cost at each node where CBOLS is stable and increases control overhead while network density grows. But HRLS average overhead cost is showing worst performance. However, it is stable between CBOLS and GLS and the result shows that they are similar as the number of nodes increases. But still CBOLS perform better in comparison with GLS.

Figure 7 shows the number of packet transmissions for each location request over the number of location requests answered as speed increases (We include the “hello” packets, to determine the number of packet transmissions). The number of packets transmitted in CBOLS is equivalent to the number of location, location request, and location response packets transmitted. Whereas the number of packets transmitted in GLS and HRLS consists of only the number of location server update packets and location server query packets. However, performance wise the CBOLS is quite constant compared to GLS and HRLS which show poor performance.

Figure 8 illustrates the effect of the node movement speed on the CBOLS location query success rate for 500 nodes. As nodes move faster, the clusters of the node are more likely to be out of date. On the other hand, the nodes

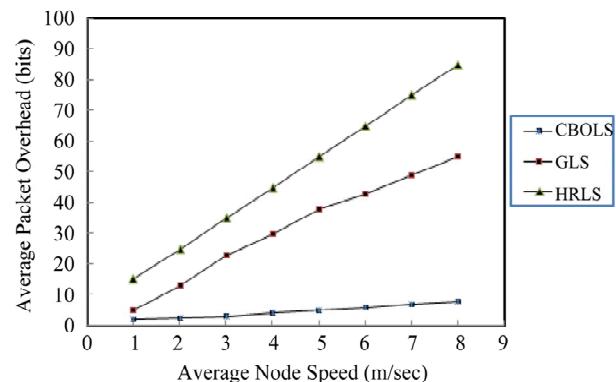


Figure 7. Average packet overhead at average speed of the node (m/sec).

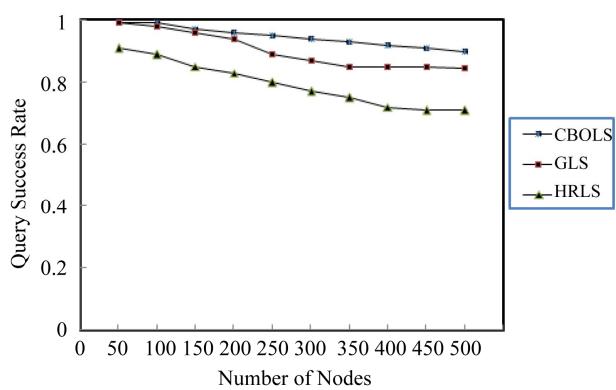


Figure 8. Average query success rate for 500 nodes in the network.

also generate updates faster. The effect of the query success rate is relatively insensitive to the node speed. The graph shows CBOLS and GLS have relatively similar results whereas HRLS is very poor in its performance.

Figure 9 plots the average end-to-end delay on a response to a location request versus number of nodes in the network. This figure indicates that the HRLS and GLS have largest percentage of location request delay provided by the requesting nodes. As location request increases, and its control overhead also increased. CBOLS have the lowest percentage right away.

The **Figure 10** shows the average end-to-end delay with regard to the increase in the function of nodes. It depicts that, a slight increase is inevitable as node-count increases in CBOLS and GLS due location updates generated by number of nodes, as it has to traverse a long-distance which will cost the overhead on clusters, whereas HRLS performs poorly.

The **Figure 11** shows that the average end-to-end delay shows slight increase in our scheme with node speed, while the other two schemes are above the normal level. However, GLS is still stable after a certain point, but HRLS shows worst performance.

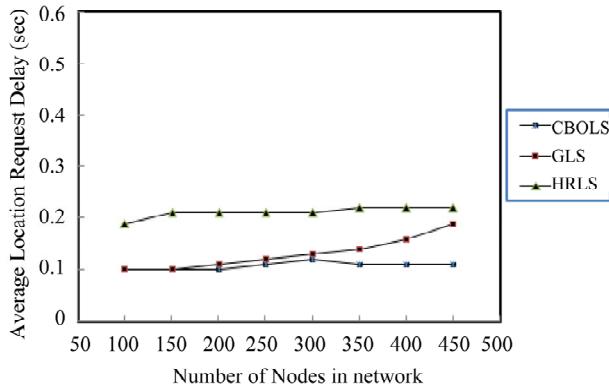


Figure 9. Average location request delay vs number of nodes.

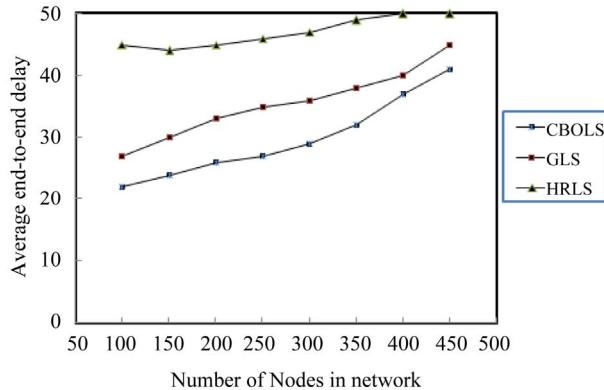


Figure 10. Average end-to-end delay as a function of increasing node-density.

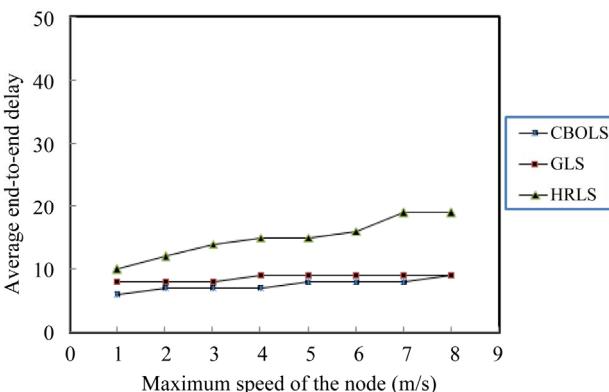


Figure 11. Average end-to-end delay (seconds) as a function of increasing node-speed.

Figure 12 depicts the normalized throughput as a function of increasing the number of nodes at different traffic scenarios. The normalized throughput is defined as the total number of packets actually delivered to their respective destinations divided by the total number of packets generated within the whole network. The throughput in GLS and HRLS scheme is higher than that of

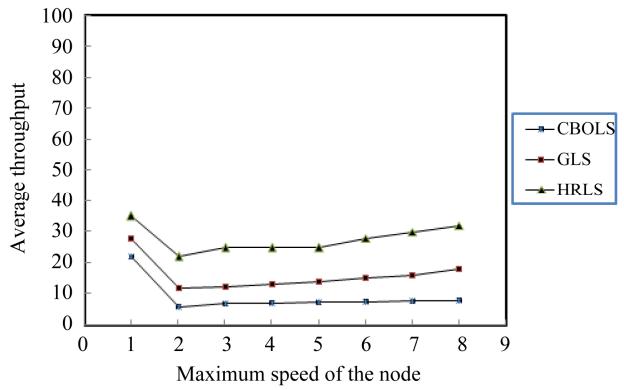


Figure 12. The average throughput vs number of nodes in the network.

CBOLS; the throughputs in both schemes tend to increase as the number of nodes increases.

Figure 13 shows the average throughput of the network with node speed increasing from 0 to 10 m/s, with pause-time 5 seconds. The throughput performance is impaired by increasing node-speed, but the extent of its effect in our scheme is very low. However, when the node density reaches a certain threshold, the normalized throughput of the network tends to drop gradually because of the IEEE 802.11 wireless channel that has fixed capacity of 2 Mbps, which is saturated due to collisions when the node-density exceeds a threshold.

6. OLC Algorithm

- 1) set Transmission range
- 2) set all nodes to “initial state”
- 3) while ClusterHead (CH) is selected do
- 4) define HelloMsg;
- 5) broadcast HelloMsg;
- 6) if (HelloMsg received by initial node = true) do
- 7) Start Timer();
- 8) call computeWeight();
- 9) if (initial node weight<=other node) do
- 10) select as CH;
- 11) return CH;
- 12) else do{
- 13) select as member;
- 14) return member;
- 15) else if (a node received HelloMsg by more than two CH)
- 16) select as Gateway (GW);
- 17) repeat step 6 - 16;
- 18) end loop;}
- 19) int broadcast HelloMsg (){
- 20) broadcast HelloMsg to all nodes;
- 21) return HelloMsg;}
- 22) int calculate Neighborsnode(){
- 23) Dc = Nodes within transmission range;

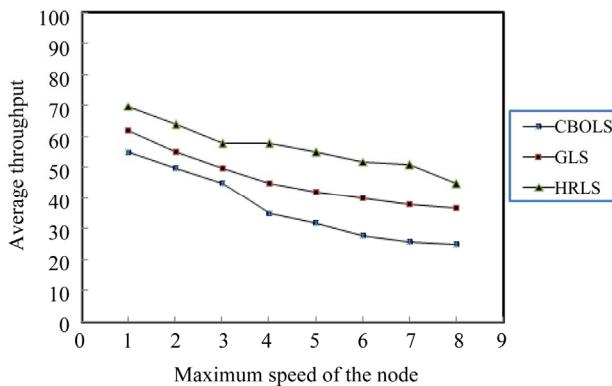


Figure 13. Average throughput as a function of increasing node-speed.

```

24) return Dc; }
25) int compute Mobility(){
26) Mb = Mb = 10log10 (RxTx) new X -> Y/(RxTx) old
   X->Y;
27) return Mb; }
28) int computeBr(){
29) Bn = Fn/Rn (t);
30) return Bn; }
31) int compute_Weight(){
32) call calculate Neighborsnode();
33) call compute Mobility();
34) call computeBr();
35) Wc = (W1 × Dc) + (W2 × Mb) + (W3 × Br) + (W4
   × Mr) + (W5 × Ps);
36) return Wc; }
37) Exit Application();
```

Performance Analysis of OLCA

We have used the network region of $2000 \times 2000 \text{ m}^2$. Nodes follow Random waypoint mobility model and connections are established between nodes using CBR (Constant Bit Rate) traffic. Nodes pause for a few seconds and then move to a randomly chosen location at a fixed speed of 10 m/s. We consider 50 nodes in the constant area and vary the node density by increasing the number of nodes from 100 to 500.

Figure 14 shows that the network size of 500 nodes with average cluster formation property. Using Lowest-ID, more number of clusters formed, and thus degraded performance of the network. Considering the routing tables in the network, this proves that Lowest-ID has limitations.

As shown in **Figure 15**, at Mobility speed of 10 m/s the total number rate of cluster re-affiliation is slowed down with OLCA and stable in increasing while network density grows, whereas with Lowest-ID the result shows that the number of cluster changes is much higher with network density of 500 nodes. However, at initial and

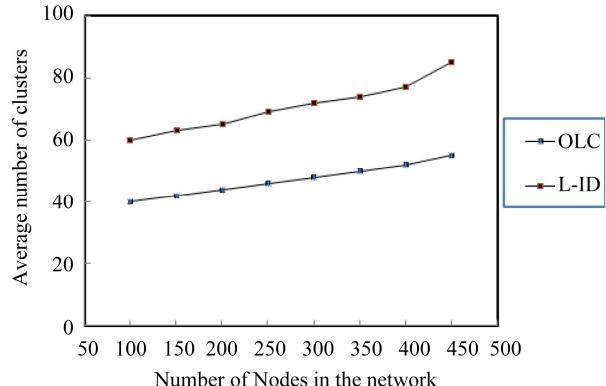


Figure 14. Total number of cluster formation with lowest-ID and OLCA using the network density of 500 nodes at mobility speed of 10 m/s.

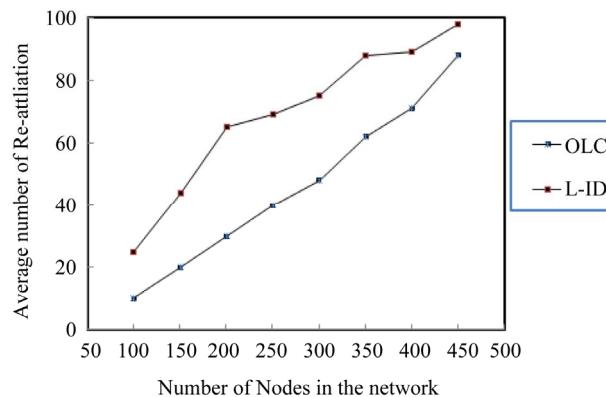


Figure 15. Total number of cluster re-election using network density of 500 nodes at mobility speed of 10 m/s.

end point the Lowest-ID also shows that the rate of cluster formation is lower and equally similar graph with our OLCA as shown in **Figure 15**.

Figure 16 shows the average number of cluster head re-election at Mobility speed of 10 m/s with network density of 500 nodes, this metric result shows our OLCA has tremendous advantage over Lowest-ID as it has very less number of cluster head re-elections happening at the mobility rate as defined 10 m/s by us. Our OLCA has been able to make less number of cluster head re-elections and thus the network is stable and robust in terms of cluster heads which can hold their status for a long time causing greater improvement in network throughput.

Figure 17 shows the results of end-to-end throughput of CBR traffics. OLCA gave lower throughput as the node density and mobility increased. Regardless of the number of nodes and speed, OLCA gave consistently better end-to-end throughput in comparison with L-ID.

Figure 18 shows the overhead of packets generated per node during the initial clustering set up phase. In L-ID the overhead is increased with node speed and the number of nodes. On the other hand, the proposed OLCA produced

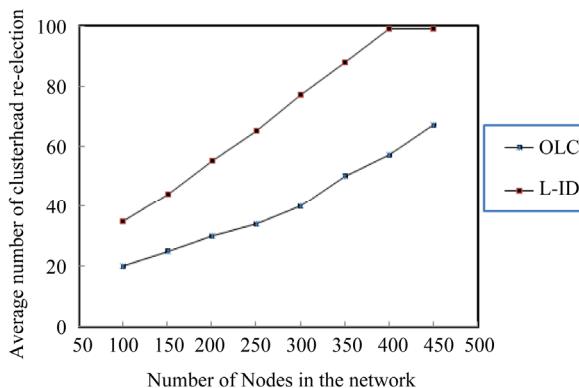


Figure 16. Total number of cluster head re-election using network density of 500 nodes at mobility speed of 10 m/s.

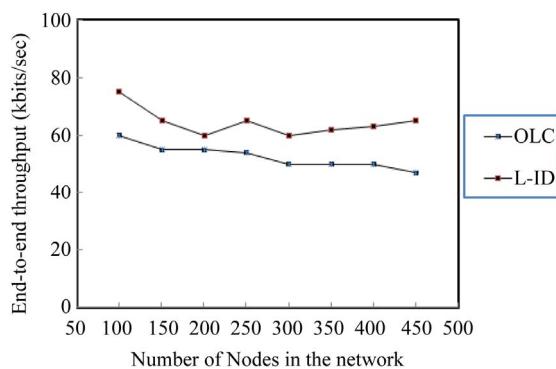


Figure 17. End-to-end throughput in OLC and L-ID.

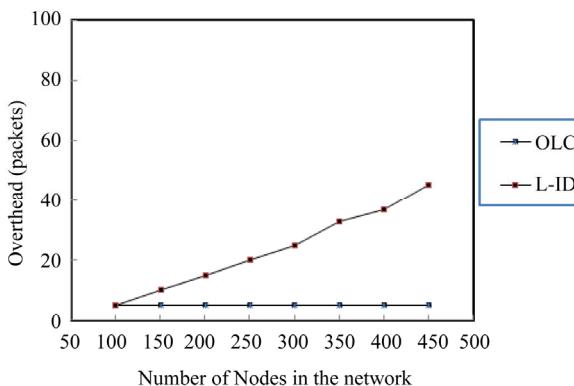


Figure 18. Overhead packets are generated during clustering setup phase.

stable and consistent overhead regardless of speed and density of the network. Such result can be obtained, since our algorithm uses neighbor information table to form clusters and eliminates delivery of packets with weight information of nodes that are not within transmission range.

7. Conclusion

The overall performance of the CBOLS is much more

effective and efficient for object location services compared with the GLS and HRLS. It is a technology which is cost efficient and time-saving with easy connectivity which can be very useful for the location services. We are also working on possibilities of comparing it with more tools and technologies in order to obtain better results, in order to make it more effectual.

REFERENCES

- [1] C. K. Toh, "Ad Hoc Mobile Wireless Networks: Protocols & Systems," 2nd Edition, Prentice Hall, Inc., Upper Saddle River, 2002.
- [2] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," SpringerLink, 2007, pp. 153-181.
- [3] C. E. Perkins and E. M. Royer, "Ad-Hoc On-Demand Distance Vector Routing," *Workshop on Mobile Computing Systems and Applications*, 1999, p. 90.
- [4] N. Beijar, "Zone Routing Protocol." www.netlab.tkk.fi/opetus/s38030/k02/Papers/08-Niclas.pdf
- [5] Y.-B. Ko and N. H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," *Wireless Networks*, Vol. 6, No. 4, 2000, pp. 307-321. [doi:10.1023/A:1019106118419](https://doi.org/10.1023/A:1019106118419)
- [6] S. Basagni, I. Chlamtac, V. R. Syrotiuk and B. A. Woodward, "A Distance Routing Effect Algorithm for Mobility," ACM Digital Library, 1998, pp. 76-84.
- [7] T. Camp, J. Boleng and L. Wilcox, "Location Information Services in Mobile Ad Hoc Networks," *Proceedings of the IEEE International Conference on Communications*, 2002, pp. 3318-3324.
- [8] M. Kasemann, H. Fubler, H. Hartenstein and M. Mauve, "A Reactive Location Service for Mobile Ad Hoc Networks," Report, 2002.
- [9] N. K. Guba and T. Camp, "GLS: A Location Service for an Ad Hoc Network," <http://toilers.mines.edu>
- [10] G. Owen and M. Adda, "SOLS: Self Organizing Distributed Location Server for Wireless Ad Hoc Networks," *International Journal of Computer Networks & Communications*, Vol. 1, No. 1, 2009.
- [11] I. Stojmenovic, "Location Updates for Efficient Routing in Ad Hoc Networks," *Handbook of Wireless Networks and Mobile Computing*, John Wiley & Sons, Inc., New York, 2002.
- [12] R. Jain, A. Puri and R. Sengupta, "Geographical Routing Using Partial Information for Wireless Ad Hoc Networks," *IEEE Personal Communications*, February 2001.
- [13] Garmin, "GPS Beginner's Guide," July 2008.
- [14] J. Nuevo, "Comprehensible GloMoSim Tutorial," 2004. INRS—Université du Québec Nuevo@inrstelecom. Uqu-ebec.ca
- [15] R. A. Meyer and R. Bagrodia, "PARSEC User Manual for PARSEC," 1999. <http://pcl.cs.ucla.edu/>