

Design and Analysis of a Multi-level Location Information Based Routing Scheme for Mobile Ad hoc Networks

Koushik Majumder¹, Sudhabindu Ray² and Subir Kumar Sarkar²

*¹Department of Computer Science and Engineering,
West Bengal University of Technology, Kolkata*

*²Department of Electronics and Telecommunication Engineering,
Jadavpur University, Kolkata
India*

1. Introduction

Classical routing algorithms for MANET are basically route based, i.e. nodes maintain routes to the other nodes in the network. Many existing routing protocols (DSDV (Perkins & Bhagwat, 1994), WRP (Murthy & Garcia-Luna-Aceves, 1996), FSR (Haas & Pearlman, 1998), ANDMAR (Gerla et al., 2000), DSR (Johnson & Maltz, 1996), AODV (Perkins & Royer, 1999), TORA (Park & Corson, 1997)) proposed within the MANET working group of IETF, are designed. These algorithms are basically of two types - proactive and reactive.

In case of proactive protocols like DSDV (Perkins & Bhagwat, 1994), CGSR (Chiang et al., 1997), STAR (Garcia-Luna-Aceves & Spohn, 1999), OLSR (Clausen et al., 2001), HSR (Iwata et al., 1999), GSR (Chen & Gerla, 1998) the nodes in the adhoc network must keep track of all the routes to all other nodes, so that, whenever a node wants to send a data packet to another destination node, it can do that without wasting any time for path setup. This necessitates periodic exchange of routing information between the nodes of the network. The immediate disadvantage of these schemes is that too much network traffic will be consumed when the size of the network or the mobility of nodes increases.

In case of reactive routing protocols such as DSR, AODV, ABR (Toh, 1997), SSA (Dube et al., 1997), FORP (Su & Gerla, 1999), PLBR (Sisodia et al., 2002) a lazy approach is applied. Here the nodes need not maintain the routes to all other nodes. Thus, there is no need of periodic exchange of routing information between nodes. Routes to the destinations are determined on demand by flooding the whole network with route query packets. The immediate disadvantage of this approach is - flooding becomes prohibitive as the size of the network grows.

Some proposed algorithms claim to have the best of these two classes. Protocols like CEDAR (Sinha et al., 1999), ZRP (Haas, 1997), and ZHLS (Joa-Ng & Lu, 1999) combine both a proactive and a reactive approach.

A new family of routing algorithms, which are known as position-based routing algorithms such as GLS (Li et al., 2000), SLURP (Seung-Chul. et al., 2001), SLALoM (Cheng et al., 2002), DLM (Xue et al., 2001), were, introduced which use information about the physical position

of the participating nodes. They eliminate some of the limitations of topology based routing algorithms by using this extra information.

Commonly, each node determines its own position using GPS or some other type of positioning service. Position-based routing protocols have certain advantages over Topology-based routing protocols.

1. The nodes have neither to store routing tables nor to transmit messages to keep routing tables up to date.
2. Reduced overhead, as the establishment and maintenance of routes is (usually) not required in a protocol that uses location information for routing.

2. Location services

A Location service is responsible for providing location information of nodes in the network. Mobile nodes register their current location with this service. When a node does not know the position of the destination node, it contacts the location server and requests that information.

Existing location services (Amouris et al., 1999) can be classified according to how many nodes host the service. This can be either some specific nodes or all nodes on the network. Furthermore, each location server may maintain the position of some specific or all nodes in the network. These can be abbreviated as:

- some-for-some
- some-for-all
- all-for-some
- all-for-all

3. Review of previous work

Several location service schemes have been proposed in the literature: GLS, SLURP, SLALoM and DLM are some representative examples.

3.1 GLS (Li et al., 2000)

Grid Location Service (GLS) divides an area containing the ad hoc network into a hierarchical grid of squares. The largest square is called the level-H square. The level-H square is then recursively divided into four level-(H-1) squares until level-0 squares are reached, forming a so-called quad-tree. In each level-i square (for $i > 0$), node A selects three location servers, one in each level-(i-1) square that A is not in. The structure of GLS is shown in Fig. 1.

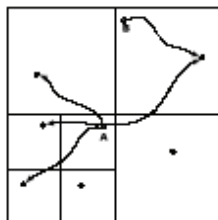


Fig. 1. Structure of GLS

GLS selects the location servers based on the node ID in each server's service area (i.e., a quadrant). For a node C to be node A's location server, C must have the smallest ID that is larger than A's ID in that quadrant, i.e., $C = \min \{x \mid \text{node } x \text{ is in the quadrant, } ID(x) > ID(A)\}$. Each node updates its location servers with its *exact* location after it moves a threshold distance δ . To query for a particular node A, a node B sends the query to the node that is closest to A for which B has location information, and so on. Eventually the query would reach one of A's location servers.

3.2 SLURP (Seung-Chul. et al., 2001)

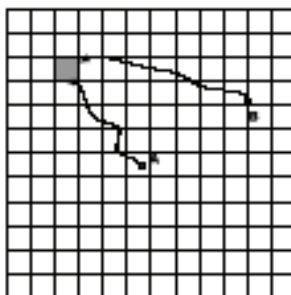


Fig. 2. Flat Grid of Squares used in SLURP

In SLURP, the entire network area (a square) is divided into a flat grid of squares. Node A selects its location servers by applying a hash function to A's ID and obtains the (x, y) coordinate of a point in the entire area. The square containing that point is called the home square for node A. All nodes in that square store A's exact location information. Every time node A moves to a different square, it updates its home square with new location information. For any node B, that wishes to communicate with node A, the same hash function is applied to node A's ID to obtain A's home square. A query packet is then forwarded to A's home square to retrieve A's location information. This is illustrated in Fig. 2.

3.3 SLALoM (Cheng et al., 2002)

SLALoM combines the strengths of SLURP and GLS. In this scheme, each node is assigned multiple home regions distributed uniformly over the area in which the nodes move about. (The nodes in these home regions act as location servers for the node.) It is assumed that the mobile nodes are capable of knowing their current location, using for example, the Global Positioning System (GPS), and are equipped with radios. It is also assumed that the nodes move about in a square region of area A. According to SLALoM, the square is divided into G unit regions called *order-1 squares*. It then combines K^2 of the order-1 squares to form *order-2 squares*. A node's home region will consist of an order-1 square. With some exceptions, every node has a home region in each order-2 square. Hence, every node has $O(A/K^2)$ home regions.

Maintaining location. Let v be a node in the network. Suppose it lies in the order-1 square R_i and R_i is inside order-2 square Q_j . We say that a home region of v is *near* v if the home region lies in Q_j or it lies in one of the eight order-2 squares that are neighbors of Q_j . Otherwise, a home region is *far* from v .

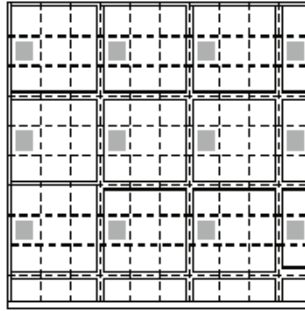


Fig. 3. Network Hierarchy of SLALoM

The following invariant always holds for v : all home regions of v know v is in Q_j . In addition, all home regions near v know v is in R_i .

Location Updation. Each time a node moves into a new order-1 square, it has to inform its 9 nearby home regions of its current exact location. This entails 9 broadcasts in a unit region. Furthermore, if such a move also causes the node to move into a new order-2 square, then it has to inform all its far home regions of its current approximate location. This requires $O(A/K^2)$ broadcasts in a unit region.

Paging. If a node u wishes to find the location of another node v , it sends a unicast to a home region of v closest to it. If this home region is near v then u obtains the exact location of v . On the other hand, if the home region is far from v then u obtains an approximate location of v . Node u then routes its message to a home region near v , R_k . The node that receives the message at R_k then sends it to the exact location of v .

3.4 DLM (Xue et al., 2001)

DLM partitions the entire network much like GLS, i.e., there are $H + 1$ level of squares. The location servers are duplicated uniformly across the region, one server in every level- K square. Here K is a system parameter between 1 and H . The servers are chosen by hashing to a point in each level- K square; therefore, we say DLM also uses a *two-level server structure*.

DLM uses two addressing policies: complete and partial address. In complete address policy, all the location servers store the exact location of a node. In case of the partial address policy, each location server stores location information with different granularity. For $i > K$, if the location server of node A is located in the same level- i square in which A resides in, the servers store only which level- $(i-1)$ square A is in. If the server is located in the same level- K square as A , the complete location information is stored.

The query operation is straightforward if the complete address policy is used. Node B simply queries the nearest location server of A to obtain A 's location. If the partial address policy is used, node B simply queries the nearest location server of A . If the complete address of A is found, then the query is complete. Otherwise the server of A indicates which level- $(i-1)$ square A is in, the query is then forwarded to A 's location server in that level- $(i-1)$ square. This process continues until A 's complete information is found.

4. Proposed scheme

Position based routing protocols need not store the route information. Here the main component is the geographic location information of the nodes. In our proposed Layered

Square Location Management (LSLM) scheme, we have assumed that each node is equipped with GPS system through which the node can acquire its current geographic location. We also assume that each node has a transmission range of r_t .

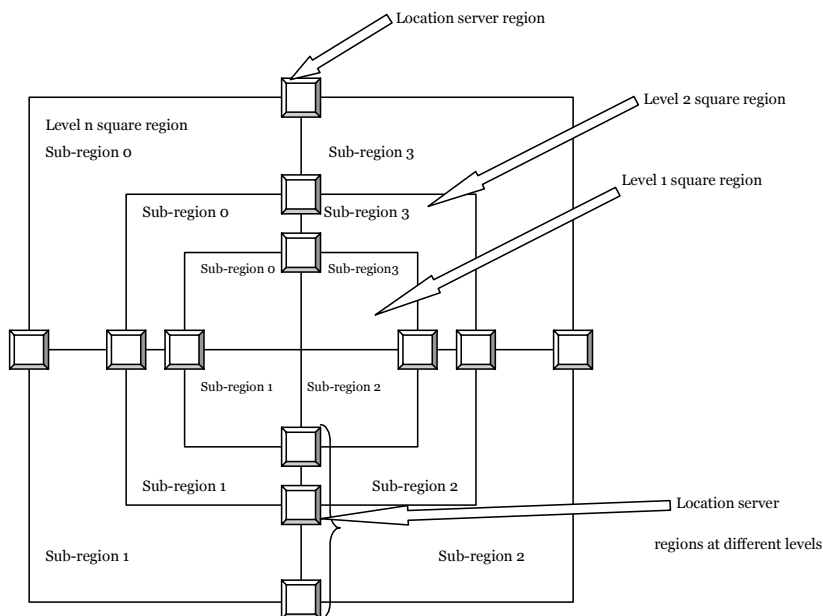


Fig. 4. Complete network structure of proposed LSLM

In our scheme, we have divided the entire network area into L level of square regions. The arrangement is such that each level i square region encapsulates the level $(i-1)$ square region and is encapsulated by level $(i+1)$ square region. Each square region has a side length of $2.2^i s$, where i denotes the level number and s depends on the node density. The innermost region is the level-1 square region and the outermost region is the level- L square region.

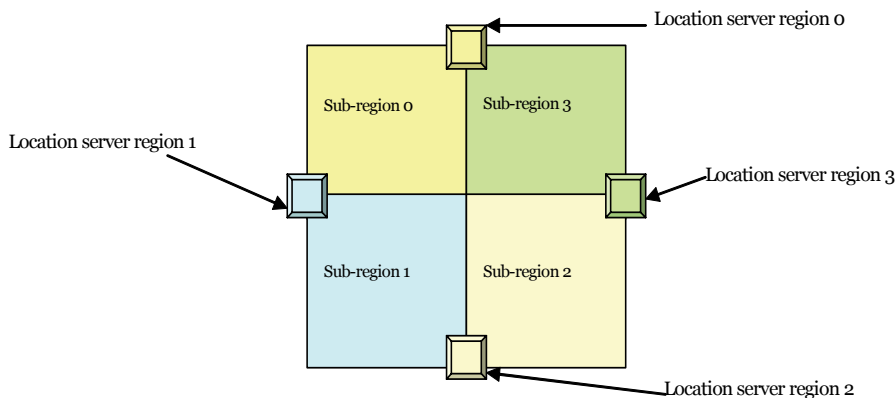


Fig. 5. Assignment of location server region

The square region at each level is further subdivided into four sub-regions: sub-region-0, sub-region-1, sub-region-2, and sub-region-3. In each level we have four location server regions, where each location server region is a square area having side length of r . All the nodes residing in the location server region act as location servers. These location servers are responsible for keeping track of the location information of the nodes. We have shown the arrangement of the location server regions within the square region at each level in Fig. 5. Each location server region has a fixed sub-region within the square region at each level assigned to it. The location server is responsible for keeping track of the location information of all the nodes within this sub-region. The Table:1 shows the assignment of the sub-regions within the square region at each level.

Location server region 0	Sub-region0
Location server region 1	Sub-region1
Location server region 2	Sub-region2
Location server region 3	Sub-region3

Table 1. Assignment of sub-regions to location server regions

In case of MANET, changes in network topology can be frequent and mobility of the nodes can be high. Therefore, cost for location update will be a major burden. If we keep track of only the exact location information of the nodes, then there is a possibility of this information becoming stale quickly as the mobile nodes frequently change their location. This will require frequent invocation of expensive location update routines. To address the issue, we have applied the concept of multi-level location information. We have assumed that the location information can be of two types – fully qualified location information and relative location information.

The fully qualified location information of a node includes the following components.

Node -id
x-coordinate
y- coordinate
Location server-id

Fig. 6. Fully qualified location information

Location server id has three components.

Level no.
Sub-region no.
Server-id

Fig. 7. Location server id

From Fig. 6 and Fig. 7, we can see that the fully qualified location information of a node A contains the current x and y coordinate position of A, the node id and the id of the location server that is currently keeping track of the location information of A. Location server id has three components embedded in it. The level no. of the square where the location server is currently in is indicated by the "level no". A location server is responsible for keeping track

of the fully qualified location information of the nodes within a sub-region and the “sub-region no.” corresponds to this particular sub-region. “Server id” uniquely identifies the sever within a location server region.

On the other hand relative location information has two components.

Location server id
Node id

Fig. 8. Relative location information

The location servers within a location server region are responsible for keeping track of the fully qualified location information of only those nodes that are currently within its assigned sub-region within a particular level *i*. On the other hand, the location servers within a particular location server region also keep track of the relative location information of all the nodes that are currently within other sub-regions of the same level *i* square region. When a node moves within a particular sub-region, it needs to notify only the single location server region - that is currently in charge of that sub-region, regarding the change in its fully qualified location information. This reduces the location update cost.

4.1 Location update

We can divide the location update mechanism in three categories.

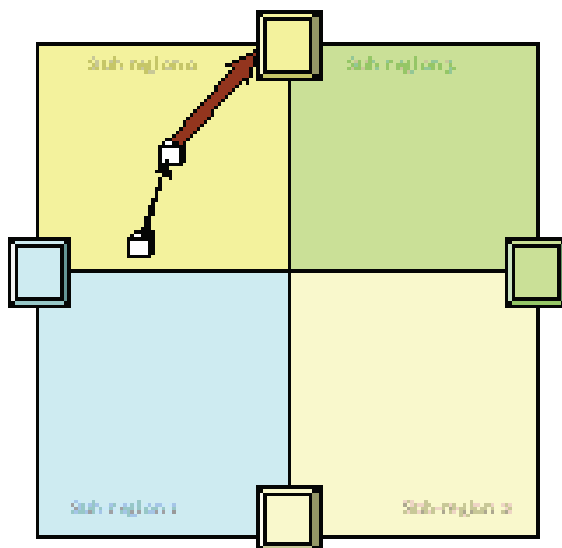


Fig. 9. Location update for node movement within sub-region

I. Location update for node movement within sub-region:

When a node A moves within its current sub-region, it needs to notify only those location servers that are currently in charge of this particular sub-region. This set of location servers are currently keeping track of the fully qualified location information of node A and any changes in the x and y coordinate positions of node A must be reflected to them. As A is

moving within its sub-region, there will be no change in its relative location information. Therefore, node A need not inform the location servers in other sub-regions of that particular level. Moreover, A does not need to notify the location server regions in other levels, as they contain neither the fully qualified location information nor the relative location information of node A.

II. Location update for node movement between sub-regions:

In this case, node A is moving from one sub-region to another within the same level i square region. After reaching the new sub-region, A probes its neighbors to get information about its new location server region. Once it gets this information, A sends its current x and y coordinate positions and the node id to the new location server region. Now node A is under the direct supervision of the new location server region. Therefore, this new location server region needs to update the location information regarding node A from relative to fully qualified one. The new location server region then needs to send the new relative location information of node A to other location server regions, which are within the same level i square region. These other location server regions now need to modify the location information about node A accordingly. If they contained the fully qualified location information, in that case, they need to update it with the new relative location information of node A. On the other hand, if they contained the old relative location information, in that case, they need to update the entry with the new relative location information of node A.

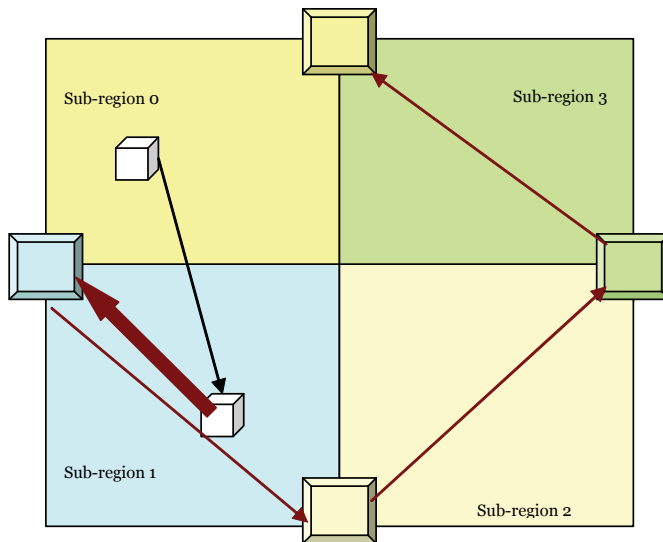


Fig. 10. Location update for node movement between sub-regions

III. Location update for node movement between square regions at different levels:

In this case, (illustrated in Fig. 11), node A moves from a square region at one level to a square region at another level. After reaching its new sub-region within its new square region, the node probes its neighbors to get information about its new location server region. Once A gets this information, it sends its previous fully qualified address and the current x and y coordinate positions to this new location server region. From node A's previous fully qualified address, the new location server region can know the previous level

no. of the node. The previous level no. is required by the new location server region in sending the new relative address of A, (i.e., current location server id and node id) to a location server region in the previous level. This information is then relayed to all the other location server regions in the previous level. Those location server regions after analyzing the current relative address of the node, find that the level no. of node A has already changed, i.e., node A is no longer in the square region at their level. Therefore, they delete the entry corresponding to node A from their database.

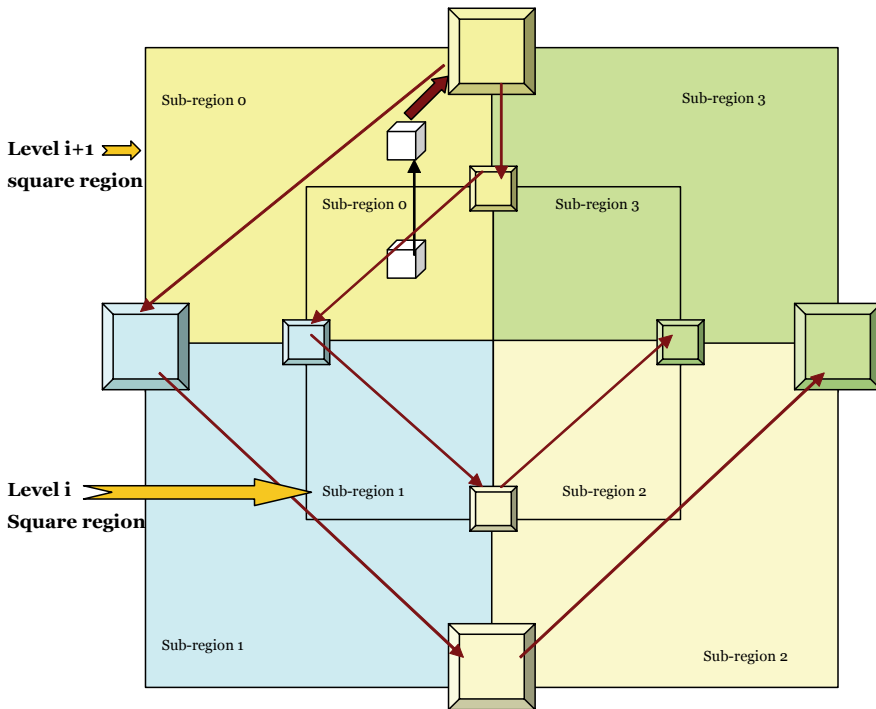


Fig. 11. Location update for node movement between square regions at different levels

The new location server region is in a square region, which is at a different level than the level of node A's previous square region. Therefore, the new location server region must make a new entry in its location information database about the new fully qualified location information of node A. This new location server region then needs to send the new relative location information of node A to other location server regions within the new square region. These other location servers previously had no location information about node A. Therefore, they need to make new entries in their location information database about the new relative location information of node A.

4.2 Location query

Suppose node S wants to send a data packet to a destination node D but the location information of node D is unknown to S. Corresponding to three location update scenarios three situations can evolve.

I. Destination D is within same sub-region at same level as of source S:

In this case the location server region that is in-charge of the sub-region contains the fully qualified address of node D. The source node S sends the data packet to the location server region. The location server region extracts the current x and y coordinate position of node D from its fully qualified address and sends the data packet to node D at that location.

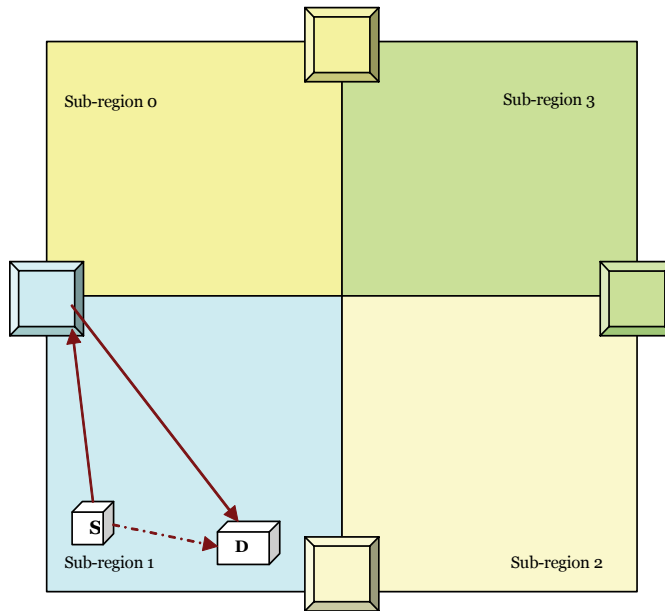


Fig. 12. Destination D is within same sub-region at same level as of source S

II. Destination D is within other sub-region at the same level as of source S:

In this case the source S sends the data packet to the assigned location server region of its sub-region. But as the destination D is within a different sub-region, therefore, the location server region of node S contains only the relative location information about destination D. From this information, the location server region of node S can find the location server region, which is currently containing the fully qualified address of node D. The location server region of node S then sends the data packet forwarded by S, to that particular location server region. This new location server region ultimately sends the data packet to the destination node D.

III. Destination D is within other square region at different level than that of source S:

The location server region now sends the data packet to the location server region of the square region that is encompassing the current level square region. It also forwards the packet to the location server region of the square region that is contained by the current level square region. The location server regions at other levels now follow the previously mentioned steps for location query. This process is continued until the destination node D is found or the network boundary is reached. Thus, if the destination node falls within the network boundary, the data packet is propagated from the source node S to the destination node D through the intermediate location server regions.

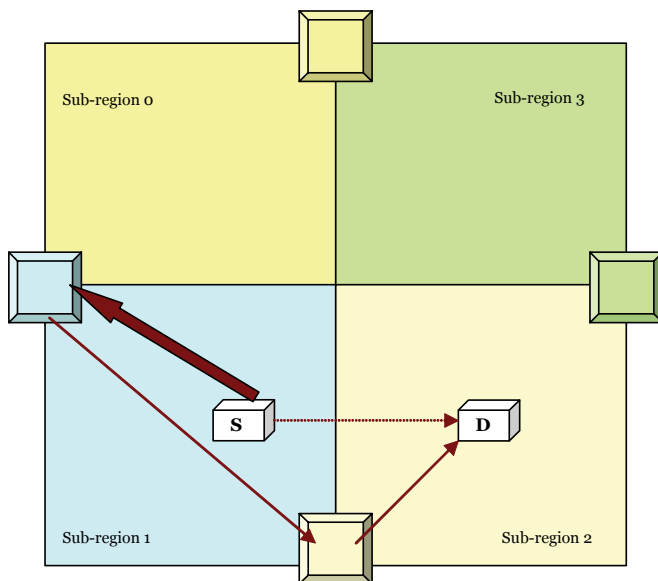


Fig. 13. Destination D is within other sub-region at the same level as of source S

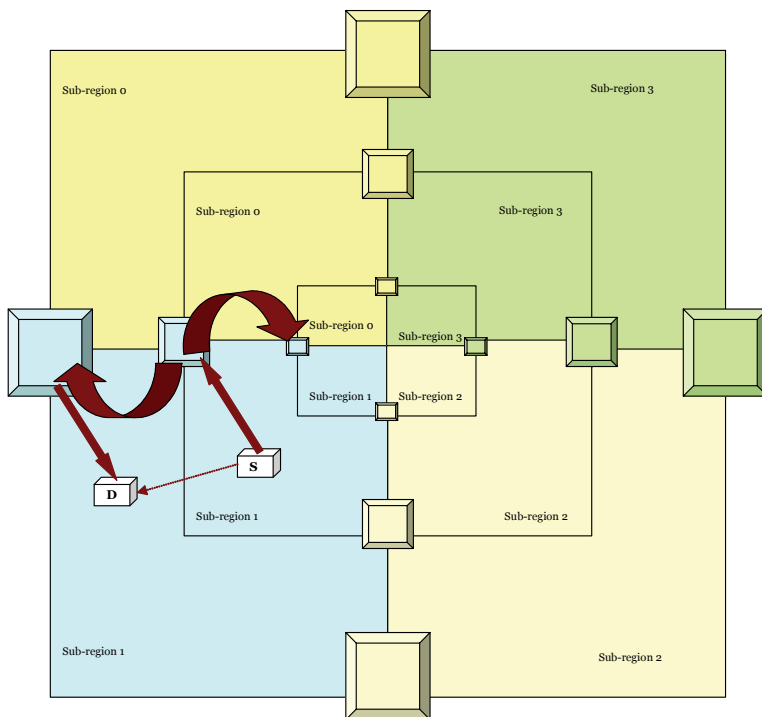


Fig. 14. Destination D is within other square region at different level than that of source S

5. Analysis of Layered Square Location Management (LSLM)

There are mainly two types of costs, which are important for any location management scheme. These are - cost for location update and cost for location query. When a node changes its position it must change its location information at the location server. The number of packet forwarding operations it needs to perform per second, in order to maintain fresh location information, is known as the location updation cost $Cost_{update}$. Similarly if a node wants to send a packet to a destination node whose location information is unknown, in that case the sender node must perform location query, to find the location information of the destination node. The number of packet forwarding operations that each node needs to perform for the purpose of location query defines the location query cost $Cost_{query}$. There is also a third type of cost, which is known as the storage cost. The storage cost $Cost_{storage}$ signifies the number of location records that each of the location servers needs to store.

In the following sections we analyze these three types of costs for our proposed Layered Square Location Management (LSLM) scheme.

5.1 Location updation cost [$Cost_{update}$]:

In our proposed scheme, location update has been divided into three parts. As a consequence, the cost for location update can also be divided into three parts - i>Cost for location update for node movement within sub-region ($Cost_{update-intra-subregion}$) ii>Cost for location update for node movement between sub-regions ($Cost_{update-inter-subregion}$) iii>Cost for location update for node movement between square regions at different levels ($Cost_{update-inter-level}$).

Thus we can write,

$$Cost_{update} = Cost_{update-intra-subregion} + Cost_{update-inter-subregion} + Cost_{update-inter-level}$$

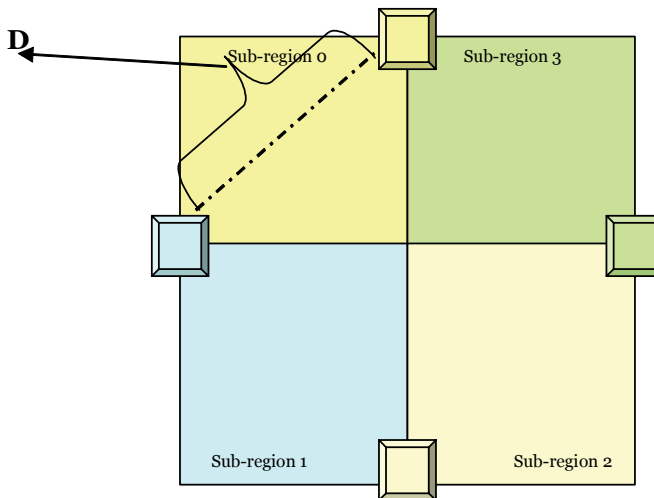


Fig. 15. Distance D

The cost for location update depends upon the amount of forwarding load, where forwarding load is determined by the number of hops traversed by a packet during location update operation. Thus the forwarding load, and as a consequence the cost will be greater for a packet traveling a greater distance. Cost for location update for node movement within sub-region ($Cost_{update-intra-subregion}$) is basically the product of updation frequency and the cost of updation of one location server region. The cost of updation of one location server region is proportional to the average number of hops an update packet takes to reach the assigned location server region. We denote this cost by Cost (1). We can approximate this cost by considering the distance $D=\sqrt{2.2^l \cdot s}$; where l denotes level number (Fig. 15).

Let us denote z as the average progress for each forwarding hop, where z is a function of the radio transmission range r_t and the node density (γ) (Seung-Chul.et al., 2001). We assume both r_t and γ are constants. Therefore, z is also a constant. It is possible to derive the average number of hops an update packet takes by D/z . If we consider the average velocity of a node as v , and the transmission range of a node as r_v , then the updation frequency is v/r_t . Thus,

$$Cost_{update-intra-subregion} = v / r_t \cdot Cost (1)$$

$$And \quad Cost (1) \propto \sum_{l=0}^L \sqrt{2.2^l \cdot s} / z$$

$$\approx \sqrt{2} \cdot s \cdot L / z.$$

If we assume S as the side length of the square region at the maximum level, i.e. L^{th} level square region, then, $S \propto 2^L$. Thus, $L \propto \log S$. Since, $S \propto \sqrt{N}$, (N =Total Number of nodes in the network), we have $L \propto \log \sqrt{N}$. Thus,

$$Cost_{update-intra-subregion} = O (v \cdot \log \sqrt{N}). \tag{1}$$

Cost for location update for node movement between sub-regions ($Cost_{update-inter-subregion}$) is the product of the boundary crossing rate (Ω) and the cost for updating the four location server regions ($Cost(4)$). So,

$$Cost_{update-inter-subregion} = \Omega \cdot Cost (4).$$

The boundary-crossing rate is proved (Yu et al., 2004) to be proportional to v . The cost of updating four location server regions can be approximated by $4(D_l)/z$. Thus

$$Cost (4) \propto \sum_{l=0}^L 4 \cdot \sqrt{2.2^l \cdot s} / z$$

$$\approx 4\sqrt{2} \cdot s \cdot L / z.$$

Therefore,

$$Cost_{update-inter-subregion} = O (v \cdot \log \sqrt{N}). \tag{2}$$

Similarly we can formulate $Cost_{update-inter-level}$ as

$$Cost_{update-inter-level} = \Omega \cdot Cost (8).$$

We can approximate the cost of updating eight location server regions by $4(D_1 + D_{1-1})/z$. Thus

$$\begin{aligned} \text{Cost (8)} &\propto \sum_{l=0}^L 6 \cdot \sqrt{2 \cdot 2^l \cdot s} / z \\ &\approx 6\sqrt{2 \cdot s \cdot L} / z. \end{aligned}$$

Therefore,

$$\text{Cost}_{\text{update-inter-level}} = O(v \cdot \log \sqrt{N}). \quad (3)$$

Thus from “(1)”, “(2)” and “(3)” we have

$$\text{Cost}_{\text{update}} = \text{Cost}_{\text{update-intra-subregion}} + \text{Cost}_{\text{update-inter-subregion}} + \text{Cost}_{\text{update-inter-level}} = O(v \cdot \log \sqrt{N}).$$

5.2 Location query cost [$\text{Cost}_{\text{query}}$]:

If a source node has some data to send to a destination node, the source node must first query a location server region to get the current location information of the destination node. The cost for this activity of querying the location information is known as location query cost ($\text{Cost}_{\text{query}}$). In order to calculate $\text{Cost}_{\text{query}}$, we have to measure the expected number of forwarding hops traveled by a query packet from the source node to its assigned location server region, which can be approximated by D/z . Therefore, the expected query cost is,

$$\begin{aligned} \text{Cost}_{\text{query}} &= \sum_{l=0}^L \sqrt{2 \cdot 2^l \cdot s} / z \\ &\propto H \\ &= O(\log \sqrt{N}). \end{aligned}$$

5.3 Storage cost [$\text{Cost}_{\text{storage}}$]:

In order to calculate the expected storage cost we need to find the average number of records stored by a location server node in the network. Dividing the total number of records stored in the network by the total number of nodes acting as location servers gives us the average number of records. Each node in the network stores its address at the four location server regions of its current layer of existence. Earlier we have mentioned that each location server region is a square area having side length of r . Hence, the area covered by a location server region can be expressed by r^2 . The average number of nodes (γ) is assumed to be constant. Thus the average number of nodes serving as location servers within a location server region is $r^2 \cdot \gamma$. Now, the expected storage cost can be expressed as

$$\text{Cost}_{\text{storage}} = (N \cdot 4 \cdot r^2 \cdot \gamma) / (L \cdot 4 \cdot r^2 \cdot \gamma) = N/L,$$

where, N = Total number of nodes in the network; L = Maximum level number. Since $L \propto \log \sqrt{N}$; the expected storage cost, $\text{Cost}_{\text{storage}} = O(N)$.

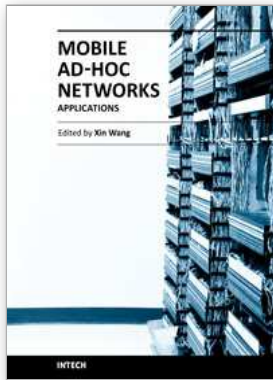
6. Conclusion

In this paper, we have presented Layered Square Location Management (LSLM), a novel scheme for the management of location information of the nodes in mobile ad hoc network. The effectiveness of a location management scheme depends on reducing the costs associated with the major location management functions- location update and location query. In case of a location service scheme we can reduce the location query cost by employing various caching strategies which is not possible for location update cost. Keeping track of only the exact location information, makes location update highly expensive due to the high mobility of nodes. In our scheme by dividing the entire network area into L levels of square regions and using multi-level location information, we have been able to provide a unique way to reduce the cost associated with both location update and location query. Further investigation on performance analysis of this scheme in different network scenarios can be taken as extended work.

7. References

- Amouris, K.N.; Papavassiliou, S. & Li, M. (1999). A position-based multi-zone routing protocol for wide area mobile ad-hoc networks. In *Proceedings of the IEEE Vehicular Technology Conference (VTC)*, pages 1365-1369
- Chen, T. W. & Gerla, M. (June 1998). Global State Routing: A New Routing Scheme for Ad Hoc Wireless Networks, *Proceedings of IEEE ICC 1998*, pp. 171-175
- Cheng, Christine. T.; Lemberg, H. L.; Philip, Sumesh. J.; Berg, E. van. den. & Zhang, T. (March 2002). SLALoM: A scalable location management scheme for large mobile ad-hoc networks. In *Proceedings of IEEE WCNC*
- Chiang, C. C.; Wu, H. K.; Liu, W. & Gerla, M. (April 1997). Routing in Clustered Multi-Hop Mobile Wireless Networks with Fading Channel, *Proceedings of IEEE SICON 1997*, pp. 197-211
- Clausen, T. H.; Hansen, G.; Christensen, L. & Behrmann, G. (September 2001). The Optimized Link State Routing Protocol, Evaluation Through Experiments and Simulation, *Proceedings of IEEE Symposium on Wireless Personal Mobile Communications 2001*
- Dube, R.; Rais, C. D.; Wang, K. Y. & Tripathi, S. K. (February 1997). Signal Stability-Based Adaptive Routing for Ad Hoc Mobile Networks, *IEEE Personal Communications Magazine*, pp. 36-45
- Garcia-Luna-Aceves, J. J. & Spohn, M. (October 1999). Source-Tree Routing in Wireless Networks, *Proceedings of IEEE ICNP 1999*, pp. 273-282
- Gerla, M.; Pei, G. & Hong, X. (August 2000). Lanmar: Landmark routing for large scale wireless ad hoc networks with group mobility. In *Proceedings of the First IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC)*
- Haas, Z. J. (October 1997). The Routing Algorithm for the Reconfigurable Wireless Networks, *Proceedings of ICUPC 1997*, vol. 2, pp. 562-566
- Haas, Z. J. & Pearlman, M. R. (August 1998). "The zone routing protocol (ZRP) for ad hoc networks (Internet Draft)"
- Iwata, A.; Chiang, C. C.; Pei, G.; Gerla, M. & Chen, T. W. (August 1999). Scalable Routing Strategies for Ad Hoc Wireless Networks, *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1369-1379

- Joa-Ng, M. & Lu, I. T. (August 1999). A Peer-to-Peer Zone-Based Two-Level Link State Routing for Mobile Ad Hoc Networks, *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1415-1425
- Johnson, David. B. & Maltz, David. A. (1996). (Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers
- Li, J.; Jannotti, J.; Couto, D. De.; Karger, D. & Morris, R. (August 2000). A scalable location service for geographic ad-hoc routing. In *Proceedings of ACM MobiCom*, pages 120-130
- Murthy, S. & Garcia-Luna-Aceves, J.J. (October 1996). An efficient Routing Protocol for Wireless Networks. *ACM Mobile Networks and App. Journal, Special Issue on Routing in Mobile Communication Networks*
- Park, V.D. & Corson, M.S. (April 1997). A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. *Proceedings of IEEE INFOCOM'97, Kobe, Japan*
- Perkins, Charles. E. & Bhagwat, Pravin. (August 1994). Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. In *Proceedings of the SIGCOMM '94 Conference on Communications Architectures, Protocols and Applications*, pages 234-244
- Perkins, Charles. & Royer, Elizabeth. (1999). Ad-hoc on-demand distance vector routing. In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*
- Seung-Chul.; Woo, M. & Singh, Suresh. (2001). Scalable routing protocol for ad hoc networks. *Wireless Networks*, 7(5):513-529
- Sinha, P.; Sivakumar, R. & Bharghavan, V. (August 1999). CEDAR: A Core Extraction Distributed Ad Hoc Routing Algorithm, *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1454-1466
- Sisodia, R. S.; Manoj, B. S. & Murthy, C. Siva. Ram. (March 2002). A Preferred Link-Based Routing Protocol for Ad Hoc Wireless Networks, *Journal of Communications and Networks*, vol. 4, no. 1, pp. 14-21
- Su, W. & Gerla, M. (December 1999). IPv6 Flow Handoff in Ad Hoc Wireless Networks Using Mobility Prediction, *Proceedings of IEEE GLOBECOM 1999*, pp. 271-275
- Toh, C. K. (March 1997). Associativity-Based Routing for Ad Hoc Mobile Networks, *Wireless Personal Communications*, vol. 4, no. 2, pp. 1-36
- Xue, Y.; Li, B. & Nahrstedt, K. (2001). A scalable location management scheme in mobile ad-hoc networks. In *Proceedings of the IEEE Conference on Local Computer Networks (LCN '01)*
- Yu, Yinzhe.; Lu, Guor-Huar. & Zhang, Zhi-Li. (2004). "Enhancing location service scalability with HIGH-GRADE," Dept. of Comp. Sci. & Eng., University of Minnesota, Technical Report TR-04-002



Mobile Ad-Hoc Networks: Applications

Edited by Prof. Xin Wang

ISBN 978-953-307-416-0

Hard cover, 514 pages

Publisher InTech

Published online 30, January, 2011

Published in print edition January, 2011

Being infrastructure-less and without central administration control, wireless ad-hoc networking is playing a more and more important role in extending the coverage of traditional wireless infrastructure (cellular networks, wireless LAN, etc). This book includes state-of the-art techniques and solutions for wireless ad-hoc networks. It focuses on the following topics in ad-hoc networks: vehicular ad-hoc networks, security and caching, TCP in ad-hoc networks and emerging applications. It is targeted to provide network engineers and researchers with design guidelines for large scale wireless ad hoc networks.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Koushik Majumder, Sudhabindu Ray and Subir Kumar Sarkar (2011). Design and Analysis of a Multi-Level Location Information Based Routing Scheme for Mobile Ad Hoc Networks, Mobile Ad-Hoc Networks: Applications, Prof. Xin Wang (Ed.), ISBN: 978-953-307-416-0, InTech, Available from: <http://www.intechopen.com/books/mobile-ad-hoc-networks-applications/design-and-analysis-of-a-multi-level-location-information-based-routing-scheme-for-mobile-ad-hoc-net>

INTECH

open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.