Binary Location-Search Based Scalable Routing Protocol For Ad Hoc Networks

Anuradha Banerjee[1] and Paramartha Dutta[2]

[1]Department of Computer Applications, Kalyani Govt. Engg. College, Nadia, West Bengal, India
anuradha79bn@gmail.com

[2]Visva-Bharati University, Santiniketan, West Bengal, India
paramartha.dutta@gmail.com

ABSTRACT

In this article, we present a binary location search based scalable routing protocol (BinLS) for ad hoc networks. It is completely distributed and based on a location management strategy that keeps the overhead of routing packets significantly small. The network is divided into some equal sized rectangular grid cells. Each node is assigned a home grid cell from where it starts its activity (transmission, reception and forwarding of messages). Nodes residing near the periphery of respective grid cells store information about all nodes present in the cell. Simulation results and detailed mathematical analysis of BinLS emphasize its merit compared to other schemes addressing the issue of scalability in ad hoc networks.

KEYWORDS

Ad hoc network, binary search, grid cell, routing, scalability.

1. INTRODUCTION

Ad hoc wireless communication is a powerful technology allowing self-organizing connectivity and network services with no pre-existing infrastructure. Due to the fact that communication is not tied to any dedicated infrastructure, ad hoc networks are potentially more resilient and pervasive. This flexibility allows the networks to be deployed where there is no place to put wiring or the cost of installing infrastructure is prohibitive [1-10]. In case of a war or disaster where temporary network connectivity is needed, ad hoc networks are extremely advantageous. Additionally, ad hoc communication allows the network to grow with the number of people using it, not requiring any infrastructure to be built.

Designing routing protocols for ad hoc networks is a challenge because of constantly changing topology triggered due to node mobility. Several authors have proposed different routing protocols for this environment [1-3] but very few of them have been evaluated with respect to their scalability. In order for large ad hoc networks to be practical, scalability problems must be solved and self-organizing distributed applications must be built. Among the existing scalable ad hoc routing protocols in literature, the mention-worthy names are : Landmark Routing [4-6], LANMAR [7], Scalable Routing Protocol [9], Hierarchical Grid Location Management (HGRID) [10], Grid Location Service (GLS) [11], Geographic Hashing Location Service (GHLS) [11], Scalable Location Management Scheme For Large Mobile Ad Hoc Networks (SLALoM) [12] and Adaptive Demand-driven Location Service [13].

Scalable routing algorithms based on coordinate or landmark hierarchies were described by Tsuchiya in [4,5,6]. Landmark nodes self-organize themselves in a hierarchy, such that landmarks at a given level of hierarchy are approximately equal number of nodes apart. Each

DOI : 10.5121/ijcnc.2010.2509

node maintains routes to nearest landmark at each level. The address of a node consists of the sequence of identifiers of the nearest landmarks, from highest to lowest levels. During routing, a node extracts from the destination address the highest landmark identifier that differs from its own address and forwards the packet towards the landmark with that identifier. The mapping from node identifier to the current address of those nodes is maintained in a distributed fashion. LANMAR [8] is a variation landmark routing that targets mobile ad hoc networks with groups of nodes that are related in function and mobility, combining some of the ideas from landmark, routing and fisheye state routing [7]. Scalability is achieved in these schemes by significantly reducing the size of per node routing table at the cost of very large routes. Increase in route length introduces delay in transmission of messages. Moreover, if any of those links break, a large number of route-requests have to be injected in the network to repair those links. This, in turn, increases the cost of messages and packet collision decreasing the packet delivery ratio of the network.

S. M Woo and Suresh Singh [9] proposed a scalable routing protocol for ad hoc networks which is also based on grid system. This protocol is based on a geographic location management strategy that keeps the overhead of routing packets very small. Nodes are assigned home regions and all nodes within a home region know the approximate location of registered nodes. As the nodes travel, they send location update messages to their home regions and this information is used to route packets. Theoretical performance results show that control packet overhead scales linearly with v_{max} and as $N\sqrt{N}$ with increasing number of nodes. The symbols v_{max} and N carry their already mentioned meaning. In HGRID [10], the network is divided into several server regions which are clustered in such a way that localized mobility causes location updates from nodes to terminate in lower order servers. Thus the distance traversed by a location update message is proportional to the level of hierarchical boundary crossing and total location update overhead has been derived to be $O(v_{max}N\log N)$. Cost of location discovery is calculated to be $O(N\sqrt{N})$.

GLS [11] is a hierarchical hashing-based location service that produces good performance when source and destination nodes are closed to one another. But, it has a very high complexity of maintaining hierarchy of grids and tracking movements of nodes across grid boundaries. GHLS [11], on the other hand, is an extreme case of flat hashing based location service that outperforms GLS in respect of both protocol overhead and success rate. Cost of locating a node scale as $O(N\log_2 N)$ while cost of location update is $O(v_{max})$. SLALoM, on the other hand, divides the network into grid squares, which are called order-1 square. K2 numbers of order-1 squares are combined to form an order-2 square. K is an integer. Its choice depends upon the protocol designer [12]. The distance a query may have to travel, is bounded by the size of any order-2 square. In case of SLALoM, overall cost scale as $O(v_{max}N^{4/3})$ while cost of location update is $O(v_{max}N)$. Significance of the symbols v_{max} and N has been mentioned earlier in this section.

ADLS [13] adopts an adaptive demand-driven approach in creating and maintaining multiple virtual home regions to improve resource-efficiency of SLALoM. In ADLS, the entire geographical area of the network is divided into regions in a way similar to SLALoM scheme but each node maintains only one primary home region in the initial condition. When a node becomes aware of the querying nodes that are not in the same oeder-2 squares as its primary home region, secondary home regions can be created in the order-2 squares of the querying sources. Each secondary home region is maintained only for a limited lifetime. A serious concern for the ADLS is that query patterns of nodes cannot be anticipated. Nodes in the same order-2 square may rarely query the same destination and as a result, a waste of resources just like SLALoM is inevitable. In addition, when a query fails in the secondary home region, the proxy node then relays the query to the primary home region of the destination. The problem of traversing very long distances to discover the destination is present in ADLS as in other above-mentioned algorithms.

In this paper, we propose a binary location search based scalable routing protocol (BinLS) for ad hoc networks. It divides the network into some equal sized rectangular grid cells. Each node is assigned a home grid cell from where it starts its activity. The home grid cell is decided based on a hash function on the node identifier. Nodes on the periphery of each grid cell store information about all nodes present in the cell. Whenever a node n_s (source) tries to communicate with another node n_d (destination), n_s computes home grid cell of the destination by applying a hash function H on n_d. If n_d is not found in the grid cell specified by its home grid cell, BinLS applies an intelligent binary search technique to locate it. As far as BinLS is concerned, its complexity of location update is $O(v_{max})$. Cost of locating a node is $O(\sqrt{N}(\log_2\sqrt{N})^2)$ if mean node density ψ is greater than or equal to 1, otherwise the cost is completely independent of number of nodes in the network. The symbols N and v_{max} carry above-mentioned meaning. BinLS is especially suitable for highly dense ad hoc networks.

2. GRID STRUCTURE OF BINLS

BinLS assumes that the network is circumscribed by the smallest rectangular grid system of size L meters × B meters (L > B). Each grid cell is of size l meters × b meters ($l \geq b$). Values of L, B, l and b are intelligently chosen so that the following criteria are satisfied:

i) $l > (L / B)$
ii) $(L/l) > (B/b)$
iii) $2\sqrt{(l^2+b^2)} < (\log_2 B)$

Values of L and B can be chosen because they do not represent actual size of the network. They are dimensions of the smallest grid system circumscribing the network. Total number of grid cells in the network is denoted as g and defined in (1).

$$g = (L \times B) / (l \times b) \quad (1)$$

In this paper, we have used the terms grid cells and cells interchangeably. Grid cells have got unique identification numbers starting from 1. Identification numbers of all other cells are 1 more than the same of their previous ones. For illustration consider the grid structure of figure 1. Values of L, B, l and b are 40, 15, 10 and 5, respectively. Total 12 grid cells are there. Coordinates of the top-left and bottom-left corners of first grid cell are (0,0) and (0,5), in that order. Similarly, coordinate of top-right corner of the same cell is (10,0). Positions of top-right corners of other cells in first row are (20,0), (30,0) and (40,0), consecutively. Also note that, (0,10) and (0,15) are positions of bottom-left vertices of grid cells numbered 5 and 9. Coordinates of other vertices of all grid cells can be computed from figure 1. Numbering of rows and columns of grid cells starts from 1.

(0,0) (10,0) (0,5) 1	(20,0) 2	(30,0) 3	(40,0) 4
(0,10) 5	6	7	8
(0,15) 9	10	11	12

Figure 1: grid structure of the network

Each node is equipped with global positioning system or GPS. They are capable of identifying their own geographical location in terms of latitude and longitude. From these, identification number of the corresponding grid cell can be computed. Below we illustrate the mechanism for a node n_i with location (32, 7) w.r.t. the network of figure 1. Assuming that n_i is placed in the grid cell at the intersection of R'-th row and C'-th column in the network of figure 1, values of R' and C' can be computed as follows:

$C' = (\lfloor 32/l \rfloor + 1) = (\lfloor 32/10 \rfloor + 1) = 4$
$R' = (\lfloor 7/b \rfloor + 1) = (\lfloor 7/5 \rfloor + 1) = 2$

Hence, identification number G of the grid cell containing n_i is given by, $G = (L / l)(R' - 1) + C' = 4(2 - 1) + 4 = 8$

For simplicity, we have assumed that nodes do not stay on and travel along the grid boundaries. But, a node may cross a grid boundary through any vertex of the associated grid cell. Minimum and maximum radio-ranges of nodes in the network are given by r_{min} and r_{max} respectively. These are also measured in meters. Please note that $b > r_{max}$. Home grid cell of a node n_i is given by H(i) where H is a good hash function. A good hash function in this context is one that uniformly distributes grid cells to nodes as home grid cells, as much as possible. Each node sends location update to its home grid cell at an interval τ. BinLS imposes an important restriction on movements of nodes. If distance of a node from its home grid cell is more than $v_{max}\tau$, then its home grid cell is changed. The node itself chooses its new home grid cell. The cell is any arbitrary cell whose center is at a distance less than $v_{max}\tau$ from the node. It is then responsibility of the node itself to broadcast identification number its new home grid cell throughout the network.

Definition 1: Periphery of a grid cell

Periphery of a grid cell consists of four non-intersecting rectangular regions s.t. all are r_{min} meters wide, with two of them having length l meters and other two having length ($b - 2r_{min}$) meters. It is illustrated in figure 2. Hashed portion denotes periphery of the cell.

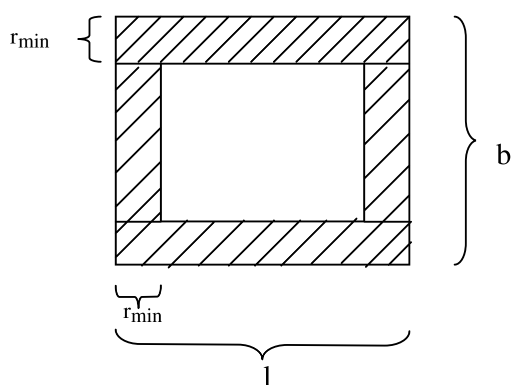

Figure 2: Periphery of a grid cell

Periphery of a grid cell consists of four non-intersecting rectangular regions s.t. all are r_{min} meters wide, with two of them having length l meters and other two having length ($b - 2r_{min}$) meters. It is illustrated in figure 2. Hashed portion denotes periphery of the cell.

Please note that area A of the periphery in figure 2 is given by,

$$A = (2lr_{min} + (b - 2r_{min})\, 2r_{min}) \tag{2}$$

Definition 2: Peripheral node

A node n_i identifies ownself as peripheral node w.r.t. a grid cell G at time t provided it entered periphery of G on or before time ($t - 2lr_{min} - (b - 2r_{min})\, 2r_{min}$) and did not leave the periphery till time t.

2.1. Locating a node in BinLS

We have already mentioned in earlier section that each node sends location update to its home grid cell at an interval τ. As soon as a node crosses a (left/right/top/bottom) boundary of its grid cell, it transmits the information in a very compact form to periphery of previous and new grid cell. For example, consider grid structure of the network in figure 2. Its top left corner is (0,0), L = 48, B = 18, l = 4 and b = 3. Let the current timestamp t be 200. Figure 3 demonstrates the movements of a node n_j between timestamps 100 and 200. The node starts from cell 30 and eventually goes to cell 40 through the cells 42, 54, 53 and 52. Identification numbers of the cells entered or left along with corresponding boundaries and timestamps are shown in table I.

First column in table 1 denotes unique identifiers of the grid cells whose boundary has been crossed by n_j. The next field entry/exit status is actually an ordered pair. Its first element can take only 2 values, 0 and 1. It is set to 0 provided n_j has exited from the associated cell, otherwise it is 1. Second element of the field entry/exit status can take 8 values starting from 0 to 7. It indicates the type of boundary / boundaries crossed. The values 0, 1, 2 and 3 correspond to the left, right, top and bottom boundaries. Values from 5 to 7 denote corners or vertices of the associated cell. Left top, left bottom, right top and right bottom corners are represented by 5, 6, 7 and 8 respectively.

1	2	3	4	5	6	7	8	9	10	11	12
13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72

Figure 3: Tracking of node movements in BinLS

Table 1: Movement of n_j between timestamps 100 and 200

Cell	Entry / Exit Status	Timestamp
30	(0,3)	105
42	(1,2)	108
42	(0,3)	130
54	(1,2)	130
54	(0,0)	170
53	(1,1)	171
53	(0,0)	178
52	(1,1)	178
52	(0,2)	182
40	(1,3)	185

From table 1, it is evident that n_j exited from grid cell 30 at time 105 by crossing its bottom boundary. Immediately before crossing the boundary, n_j geocasted the information (30,0,3,105) to periphery of cell 30. n_j discovered its first neighbor in cell 42 at time 108. Hence, the information (42,1,2,108) was transmitted to all nodes in periphery of cell 42 and so on.

Whenever a node n_i needs to communicate with another node n_j, n_i inquires any peripheral node n_k of the grid cell with identification number H(j). Let, according to n_k, last-known location of n_i is (h,q) and the corresponding timestamp is t1. As per the technique described earlier in this section, n_i computes identification number of the grid cell circumscribing the location (h,q). Then, n_i inquires any peripheral node of that grid cell. If n_j is found there, then n_i starts communicating with it directly. Otherwise, n_i issues a binary search mechanism to locate n_j. Let t be the current timestamp and v_{max} be the maximum node velocity throughout the network. Then, maximum possible amount of drift of n_j from location (h,q) within time interval (t-t1) is v_{max}(t-t1) in all directions. So, current minimum and maximum x-coordinates of n_j are MAX (1, h -v_{max}(t - t1)) and MIN (L, h + v_{max}(t - t1)), respectively. Similarly, current minimum and maximum y-coordinates of n_j are MAX (1,q -v_{max}(t - t1)) and MIN (B, q + v_{max}(t - t1)), respectively. The functions MAX and MIN both take two numbers as inputs. As the name suggests, MAX extracts the maximum number from the set of its inputs while MIN extracts the minimum.

$$\text{MAX}(c,d) = \begin{cases} c \text{ if } c > d \\ d \text{ otherwise} \end{cases}$$

$$\text{MIN}(c,d) = \begin{cases} c \text{ if } c < d \\ d \text{ otherwise} \end{cases}$$

Binary search technique is applied to first find out that particular grid cell which contains n_j. Below we discuss in details the process for computing it.

Let mx=MAX(1,q-v_{max}(t - t1)), mn=MIN (B, q+v_{max}(t - t1)), xmax=MAX(1,h -v_{max}(t-t1)) and xmin = MIN (L,h+v_{max}(t-t1)). If mn = B then mn = mn – 1. Similarly, if xmin = L then xmin = xmin – 1. Also assume that, mxrow=($\lfloor$mx/b$\rfloor$+1), mnrow=($\lfloor$mn/b$\rfloor$+1), mxcol=($\lfloor$xmax/l$\rfloor$+1) and mncol=($\lfloor$xmin/l$\rfloor$+1). Present location of n_j must be confined within portion of the network bounded by the rows mxrow, mnrow and columns mxcol and mncol. Let this fragment of the network be termed as U. Its top and bottom boundaries consist of grid cells of rows mxrow and mnrow, respectively, between columns mxcol and mncol. Similarly, left and right boundaries of U consist of grid cells of columns mxcol and mncol, respectively, between the rows mxrow and mnrow. For the first run, midrow is assigned ($\lfloor$q/b$\rfloor$+1). In all subsequent runs value of midrow is set to $\lfloor$(mxrow+mnrow)/2$\rfloor$.

n_i transmits monitor_up and monitor_down messages to all nodes in top and bottom row of grids respectively of network fragment U. All nodes that receive the monitor_up message continue to look for n_j until further notification from n_i. Their task is to inform n_i as they observe n_j crossing top boundary of U. The nodes that receive monitor_down message are supposed to inform n_i as they observe n_j crossing bottom boundary of U. Then n_i assigns any arbitrary peripheral node n_p of mxcol-th cell of row midrow, the responsibility to propagate route_request message from n_i for n_j, to any arbitrary peripheral node of each cell of row midrow of U. n_p is termed as controller node of row midrow. All nodes that receive this route_request wait for a time period of (2l+ (b -2r_{min})) before sending the reply. This time period is actually the number of transmissions required to broadcast information within periphery region of any grid cell. Reply from each node contains the largest timestamp when they last encountered n_j, provided the timestamp is less than (t-τ). Also the boundary through which n_j exited, is returned. If no such timestamp is available to a node, it keeps mum. If the reply accompanying largest timestamp indicates that top boundary of the associated grid cell of U has been crossed during exit, then mnrow is set to (midrow-1) and mxrow remains unchanged. On the other hand, if reply with largest timestamp indicates that bottom boundary of the associated grid cell of U has been crossed for exiting, then mxrow is set to (midrow+1) while mnrow remains unchanged. In both the above situations, new value of midrow is calculated as $\lfloor$(mxrow+mnrow)/2$\rfloor$. Let it be denoted as midrow′. n_p delegates its responsibility to an arbitrary peripheral node n_v of mxcol-th grid cell of row midrow′ and so on. n_v takes the role of controller in row midrow′. The procedure terminates as soon as n_j is located. The node that locates n_j, sends a message final_reply to n_i through its previous controllers informing the identification number of the present grid cell of n_j. Messages transferred between subsequent controllers are called control_delegation. If, in any stage of binary search no route_reply is obtained from any node in row midrow and at least one node in midrow of previous pass observed n_j to go up of its top boundary, then new value of mnrow is set to the value of midrow in current pass. Similarly, If, in any stage of binary search no route_reply is obtained from any node in row midrow and at least one node in midrow of previous pass watched n_j to go down crossing its bottom boundary, then new value of mxrow is set to the value of midrow in current pass.

Please note that, in each pass, both the number of rows to consider and distance between consecutive controller nodes, are becoming halved (according to the logic of binary search). In worst case, number of grid cells between two consecutive controller nodes is (B / b). If a receiver of monitor_up message observes n_j going up of the network fragment U, mxrow is set to 0 and mnrow is set to the previous non-zero value of mxrow. Similarly, If a receiver of monitor_down message observes n_j going down of the network fragment U, mxrow is set to mnrow and mnrow is set to ($\lfloor B/b \rfloor$+1). For the purpose of illustrating with an example, below we show the steps of tracking n_j according to BinLS with respect to figure 3 and table I. Please assume the followings also:

1) Home grid cell of node n_j is 18.
2) Last location update came from n_j to cell 9 at time 100.
3) Current timestamp is 200.
4) Last known location of n_j is (22,7).
5) v_{max} is 0.15 meters/sec.
6) Value of τ is 1000 seconds.

v_{max} (t-100) evaluates to 15. Minimum and maximum possible y coordinates are MAX(1,7-15) and MIN(18,7+15) i.e. 1 and 18 respectively. Hence, mxrow = 1, mnrow = 6 and midrow=3 (row containing cell 30). Similarly xmax = MAX (1,22-15) and xmin = MIN (48, 22 + 15) i.e. 7 and 37 respectively. mxcol and mncol evaluate to 2 and 10, in that order. n_i assigns any arbitrary node n_p in cell 26 (2nd cell in row 3) the responsibility to propagate route_request to one peripheral node in each cell of row 3 between columns 2 and 10 and also process their replies. It is clear from table I that, among the replies from communicated nodes in row 3, the one in cell 30 produces highest timestamp and also indicates that bottom boundary has been crossed during exit. Hence, new value of mxrow is set to (midrow+1) i.e. 4 and mnrow continues to be 6. New value of midrow is $\lfloor 4+6 \rfloor$ / 2 i.e.5. Now, n_p assigns any arbitrary node n_w in cell 50 (2nd cell in row 5) the responsibility to propagate route_request to one peripheral node in each cell of row 5 between columns 2 and 10 and also process their replies. Non-null reply is obtained from communicated peripheral nodes in cells 52, 53 and 54 of row 5. Among them, the one from cell 52 produces largest timestamp (according to table I). The corresponding entry/exit status indicates that top boundary has been crossed during exit. So, in the next run, mnrow is set to (midrow-1) i.e.4 while mxrow continues to be 4. New value of midrow evaluates to $\lfloor 4+4 \rfloor$ / 2 i.e.4. n_w now delegates the responsibility to search n_j in row 4 to any arbitrary node n_v in cell 38 (2nd cell in row 4). Among the cells in row 4, reply is obtained from cells 40 and 42. According to table I, communicated node in cell 40 produces largest timestamp and also generates the message that n_j is presently staying in it. n_v returns the reply to n_i.

2.2. Pseudocode of BinLS

Pseudocode of BinLS appears below.

Procedure node_binsrch (n_j, mxrow, mnrow, Umxrow, Umnrow, Umxcol, Umncol, monitorup, monitordowm, L)

/* Umxrow, Umnrow, Umxcol and Umncol are the top, bottom, left and right boundaries of the network fragment encompassing all possible positions of node n_j */

/*monitorup and monitordown both initialized to 0. monitorup is set to 1 if n_j goes up of Umxcol. Similarly, monitordown is set to 1 if n_j goes down of Umncol. L carries its usual meaning */

begin
pass = 0

/* pass indicates number of passes through the loop */

loop
pass = pass + 1.
If pass = 1 and monitorup = 0 and monitordown = 0 then
Begin

/* The node is remaining within the network fragment Umxrow, Umnrow, Umxcol and Umncol */
mxrow = Umxrow
mnrow = Umnrow
high = Umxrow
low = Umnrow
end
If pass = 1 and monitorup = 1 and monitordown = 0 then
Begin

/* The node has crossed the top boundary Umxrow */

mxrow = 1
mnrow = Umxrow
high = 1
low = Umxrow
end
If pass = 1 and monitorup = 0 and monitordown = 1 then
Begin

/* The node has crossed the bottom boundary Umnrow */

mxrow = Umnrow
mnrow = $\lfloor (L - 1) \rfloor / l + 1$
high = Umnrow
low = mnrow
end
exit when n_j is found or mxrow > mnrow.
midrow = $\lfloor mxrow+mnrow \rfloor /2$.
send_route_req(n_j, midrow, Umxcol, Umncol).

/* send_route_req is a procedure that sends route_request messages to all nodes of row midrow between the columns mxcol and mncol for locating n_j */

repl = receive_route_reply(n_j , midrow,Umxcol,Umncol).

/* receive_route_reply is a procedure that receives route replies corresponding to the route_requests for n_j and stores them in repl */

```
 tm = extract_timestamp(repl);

/* extract_timestamp is the procedure that extracts timestamps from repl and stores in tm */

if tm is not null then
begin
 n_k = largest_timestamp(repl, tm).
(x,y) = entry_exit(repl, n_k )

/* largest_timestamp is the procedure that accepts two arguments, repl and tm and returns the
node with largest timestamp in its route_reply. Corresponding entry or exit status is returned by
another procedure entry_exit */

 if x = 1 return n_k
 if y =2 or y = 5 or y = 7 then
 /* top boundary has been crossed */

 begin
 mxrow = high
 mnrow = midrow
 up = 1
 down = 0

/* up and down are variables which are initialized to 0. up is set to 1 if n_j has crossed top
boundary of current midrow. Similarly, down is set to 1 if n_j has crossed bottom boundary of
current midrow */

 end
 else

/* bottom boundary has been crossed */
 begin
 mxrow = midrow
 mnrow = low
 down = 1
 up = 0
 end
 end
 else
 begin

 /* tm is null */

 if up = 1 then
 mnrow = midrow
 else
 mxrow = midrow
 end
 end loop
 end
```

2.3. Complexity analysis of BinLS

Please recall from our discussion in introduction that, scalability of a protocol is best characterized by its control message overhead required to route data packets between nodes. In this section we develop a theoretical model to describe the scalability of BinLS. Specifically we show that,

- BinLS is proportional to the network-wide maximum node speed v_{max} as far as cost of location update is concerned.
- BinLS scales as $O(\sqrt{N}(\log_2 \sqrt{N})^2)$ where N is the total number of nodes in the network

As a first step in the analysis, it is necessary to identify two major components of the protocol that contribute to the message cost.

- The cost of maintaining a node's location at regular intervals – we call this the location update cost.
- The cost of locating with a node when a data packet needs to be sent to it – we call this location detection cost.

1. Location Update Cost Vy

Location Update Cost Vy is summation of two cost components – *Update Message Traversal Cost* (V_M) and *Cell Change Broadcast Cost* (V_C). Each node generates location update message at a n interval of τ for its home grid cell. Cost of traversal of this message from current grid cell to home grid cell of the respective node, is termed as *Update Message Traversal Cost* and it is denoted as V_M. Whenever a node goes to a neighboring grid cell, it broadcasts the information to periphery of both the old and new cells. Total Cost of these broadcast operations is called *Cell Change Broadcast Cost* and it is denoted as V_C. Below we discuss worst-case computation of these two differently.

1a. Update Message Traversal Cost V_M

Distance $D_i(t)$ between a node n_i and its home grid cell H(i) at time t, is less than or equal to $v_{max}\tau$ where τ is location update interval in seconds . Then, cost of messages V_M for covering distance $D_i(t)$ with hops each of size z, is given by,

$$V_M = D_i(t) / z \le (v_{max}\tau / z) \tag{3}$$

In worst case, the average one hop progress z of signal generated from any node, can be approximated as the average of maximum distance between a node with transmission radius r_{min} and its neighbors. This approximation assumes that farthest neighbor from the sender is always in the direction towards the destination. It is shown in [30] that this approximation works really well in ad hoc networks. Number of nodes ξ residing within radio-circle of a node with radius r_{min} is given by,

$$\xi = \psi \pi r^2_{min} \tag{4}$$

The probability $F(\rho)$ of all ξ nodes residing within distance ρ ($0 \le \rho \le r_{min}$) from center of the transmission circle, can be expressed as

$$F(\rho) = [\pi\rho^2 / \pi r^2_{min}]^{\xi} = \rho^{2\xi} / r^{2\xi} \tag{5}$$

The probability density function $f(\rho)$ of average one hop progress ρ is given by,

$$f(\rho) = \frac{\partial}{\partial \rho} F(\rho) = (2\xi \rho^{2\xi-1}) / r^{2\xi}_{min} \tag{6}$$

The average progress z is then the expected value of ρ w.r.t. pdf $f(\rho)$,

$$z = \int^{r_{min}} \rho f(\rho)\, d\rho = (2\xi r_{min} / (2\xi+1)) = \vartheta(\xi, r_{min}) \tag{7}$$

0

ϑ is a many-to-one monotonically increasing function.
Putting the value of z of (7) in (3) we get,

$$V_M = v_{max}\tau / \vartheta (\xi, r_{min}) \quad (8)$$

So, V_M is $O(v_{max})$.

1b. Cell Change Broadcast Cost V_C

Please refer to figure 2. Area A_p of periphery of each grid cell is given by,

$$A_p = (2lr_{min} + (b-2r_{min})\, 2r_{min}) \quad (9)$$

V_C is the number of transmissions required to cover the area A_p with circles of radius r_{min}.

$$V_C = A_p / r_{min} \quad (10)$$

Overall location update cost $Vy = V_M + V_C$
Therefore, Vy is $O(v_{max})$.

2. Location Detection Cost V_D

V_D is actually summation of number of transmissions of various messages like monitor_up, monitor_down, route_request, route_reply, control_delegation and final_reply. Among these, monitor_up and monitor_down messages are broadcasted to peripheries of at most L/l grid cells. Broadcast cost in periphery of one cell is V_C. Hence, cost V_{DM} incurred due to monitor_up and monitor_down messages is expressed as $2(L/l)(A_p/ r_{min})$. Please note that,

$$L \times B = N / \psi \quad (11)$$

Since $B < L$, $\psi \geq 1$ implies that $B \leq \sqrt{N}$. According to section II network parameters are so chosen that $(L/l) < B$. So, V_{DM} is $O(\sqrt{N})$ provided $\psi \geq 1$.

Number of route_requests in each of the communicated grid cells is equal to number of runs. Since number of rows is (B / b), total number of runs in the binary search is $\log_2(B/b)$. In worst case, controller of each row communicates with one node in all grid cells of a row. Number of grid cells in a row is (L/l). Moreover, maximum distance ς_{row} between two nodes in consecutive cells in the same row, is $\sqrt{(4l^2+b^2)}$. Average number of intermediate nodes ζ between two consecutive nodes communicated by the controller in that row, is the number of transmissions required to cover the length ς_{row} by minimum possible transmission radius r_{min} in the network. So, ζ is expressed as $(\varsigma_{row}/r_{min})$. These intermediate nodes forward route_request messages. Therefore, cost V_{DG} of route_request messages during binary search is less than or equal to $\zeta(L/l)\log_2(B/b)$. We have mentioned in section II that $2\sqrt{(l^2+b^2)} < (\log_2 B)$ is a constraint on BinLS. So, $\sqrt{(4l^2+b^2)} < (\log_2 B)$. It has been proved earlier in this section that if $\psi \geq 1$ then $B \leq \sqrt{N}$ (B/b must also be less than $\sqrt{N}$) i.e. $(\log_2 B) \leq (\log_2 \sqrt{N})$. Combining these we can write that if $\psi \geq 1$ then $\sqrt{(4l^2+b^2)}$ is $O(\log_2 \sqrt{N})$. As per the constraints enforced on BinLS (mentioned in section II), (L/l) is less than B. So, given that $\psi \geq 1$, (L/l) is $O(\sqrt{N})$ and hence $\zeta(L/l)\log_2(B/b)$ is $O(\sqrt{N}(\log_2\sqrt{N})^2)$. All nodes that receive route_request transmit route_reply. So, cost V_{DR} of route_reply is same as V_{DG}.

Cost V_{DC} of control_delegation messages consists of two components. The first one is the message sent from the sender to the first controller. Second component consists of the messages sent from a controller to the next one, starting from the second controller. Since maximum number of grids between two nodes in a row is (L/l) and average number of intermediate nodes ζ between two nodes in consecutive cells of the same row is ς_{row}/r_{min}, highest cost incurred by the first component of V_{DC} is limited to $\zeta(L/l)$ which is $O(\sqrt{N}\log_2\sqrt{N})$ in case $\psi \geq 1$ (proof is given in the earlier paragraph). We have already mentioned in earlier section that maximum number of grid cells between consecutive controller nodes is (B/b) and it becomes halved in each run. Please note here, maximum possible distance ς_{col} between two nodes in consecutive grid cells of same column is $\sqrt{(4b^2+l^2)}$. So, average number of intermediate nodes λ between two controller nodes separated by Υ grid cells is, $\Upsilon\varsigma_{col}/r_{min}$. This imposes the upper bound of V_{DC} to be, $((B/b)+(B/2b)+(B/4b)+\ldots+1)\lambda)$ i.e. $((2B/b-1)\lambda)$. Since formal reply is sent from the last controller to the sender of the communication session, cost V_{Df} of formal route is also $((2B/b - 1)\lambda)$. We have mentioned in section II that $2\sqrt{(l^2+b^2)} < (\log_2 B)$ is a constraint in BinLS and if $\psi \geq 1$ then $(\log_2 B) \leq (\log_2\sqrt{N})$. Combining these two we can write that if $\psi \geq 1$ then $\sqrt{(4b^2+l^2)}$ is $O(\log_2\sqrt{N})$.Therefore λ is $O(\log_2\sqrt{N})$. Under the same situation $B < \sqrt{N}$ and hence, B/b is $O(\sqrt{N})$. As a result it can be concluded that both V_{DC} and V_{Df} are $O(\sqrt{N}\log_2\sqrt{N})$ if $\psi \geq 1$. Location detection cost V_D is expressed in terms of its components, as follows:

$$V_D = V_{DM} + V_{DG} + V_{DR} + V_{DC} + V_{Df} \tag{12}$$

i.e. $V_D = V_{DM} + 2\,V_{DG} + 2\,V_{DC}$

If $\psi \geq 1$ then V_{DM} is $O(\sqrt{N})$, V_{DG} is $2\,O(\sqrt{N}(\log_2\sqrt{N})^2)$ and V_{DC} is $2\,O(\sqrt{N}\log_2\sqrt{N})$. So, V_D is $O(\sqrt{N}(\log_2\sqrt{N})^2)$. For values of ψ less than 1, V_D is completely independent of N.

Here we have shown that if ψ is greater than or equal to 1, then location update cost V_D is $O(\sqrt{N}(\log_2\sqrt{N})^2)$. Because $(L/l) > (B/b)$ and $2\sqrt{(l^2+b^2)} < (\log_2 B)$ as per section II, expressed as a function of network size, V_D is $O((L/l)\log_2(B/b)\log_2(B))$ i.e. completely independent of N. Analytically, this proves the supremacy of BinLS over the state-of-the-art scalable routing protocols GHLS and SLALoM.

3. SIMULATION RESULTS

The simulations use CMU's wireless extensions [14] for the ns-2 [15] simulator. Nodes use the IEEE 802.11 radio and MAC model provided by the CMU extensions; each radio's range is approximately a disc with radius 0.5-5 meters. Simulations without data traffic use 1 Megabit per second radios and those using data traffic use 2 Megabits per second radios. Each simulation runs for 10000 seconds. Each data point is presented as an average of 30 simulation runs.

The nodes are placed uniformly at random locations in a rectangular universe of size 50 meters × 20 meters in first 10 simulation runs, 70 meters × 50 meters in subsequent 10 simulation runs and 100 meters × 70 meters in last 10 runs. Size of each grid cell has been assumed to be 2 meters × 2 meters in first 10 runs, 3.5 meters × 2.5 meters in subsequent 10 runs, and 5 meters × 3.5 meters in last 10 runs. Total number of nodes is varied between 100 and 5000. Network-wide maximum speed is taken to be 0.3 meters per second. Each node moves according to a random-waypoint mobility model in first 10 simulation runs, gaussian model in subsequent 10 and random walk model in last 10 runs. A node chooses a random destination and moves toward it with a constant speed which is less than or equal to v_{max} meters per second, where v_{max} ranges from 0.05 to 0.30 in different simulation runs with the intermediate values being 0.08, 0.12, 0.16 and 0.20. When it reaches the destination, it waits for a random time interval before choosing a new destination and begins moving toward it. The simulations in this section measure the behavior of BinLS while forwarding data traffic.

The 802.11radio bandwidth is 2 Megabytes per second and upper bound of location update interval is 60 minutes. The data traffic is generated by a number of constant bit rate connections equal to half the number of nodes. No node is a source in more than three connections and destination in more than five connections. For each connection, four 128 byte data packets are sent per second. For the purpose of comparison, we include results for GHLS [11] and SLALoM [12]. We have taken *Query Success Rate, Data Packet Delivery Ratio, Cost of messages* and *Delay In Locating Destination as metrics*. These are plotted with respect to total number of nodes in the network in figures 4, 5, 6 and 7 respectively. Total number of nodes in the network has been varied from 100 to 5000 with the intermediate values being 800, 1500, 2200, 2900, 3600 and 4200 in that order.

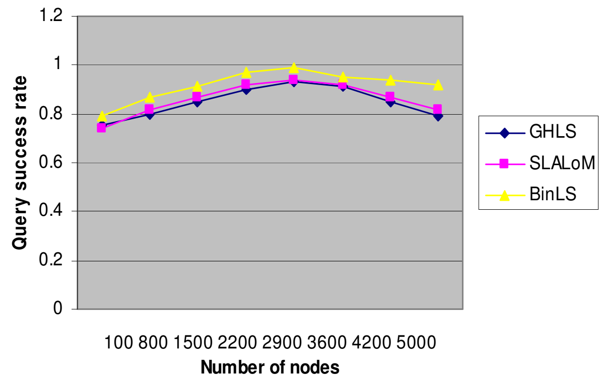

Figure 4: Query success rate vs. number of nodes

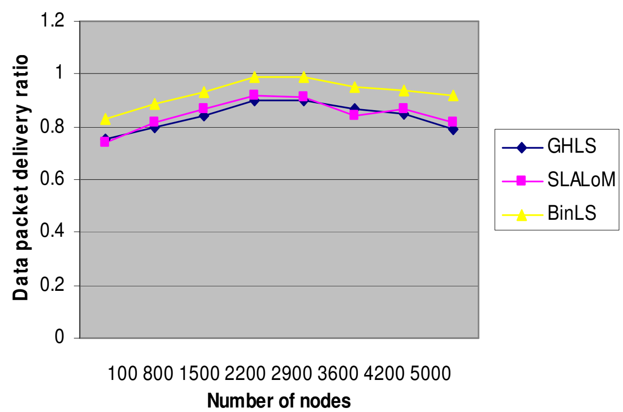

Figure 5: Data packet delivery ratio vs. number of nodes

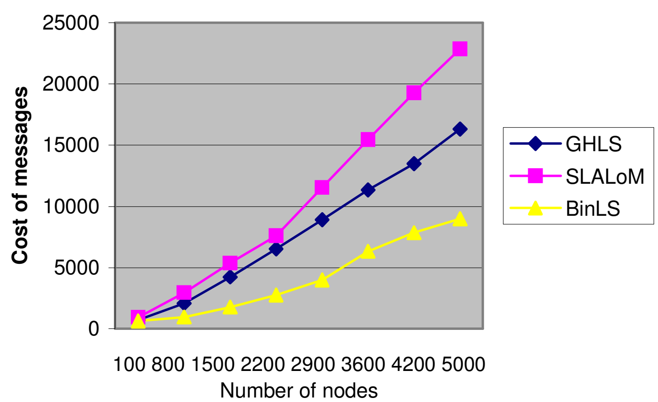

Figure 6: Cost of messages vs. number of nodes

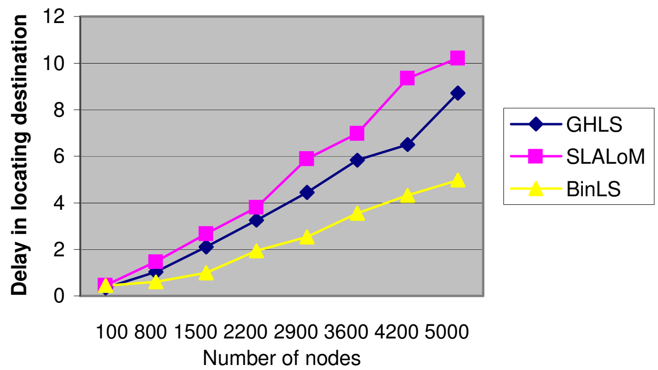

Figure 7: Delay in locating destination vs. number of nodes

Please note that the problem of holes is equally devastating for all three of GHLS, SLALoM and BinLS. Unlike GHLS and SLALoM, a source node in BinLS can correctly locate its desired destination without broadcasting route_requests throughout the network when information about the present geographical location of the destination is not available in its home grid cell. So, the resultant message cost for BinLS is much smaller than its competitors. Reduction in message cost reduces the chance of signal collision and hence, requirement of retransmission is also minimized. Indirectly it increases data packet delivery ratio. Actually data packet delivery ratio is proportional to the number of data packets transmitted by several sources unless the effect degenerates with signal collisions caused by extremely huge node density. Also notice that, as the number of nodes increase beyond 4200, rate of deterioration of performance of the system, which is dependent upon signal collision is much smaller in BinLS.

4. CONCLUSION

BinLS is a binary location search based scalable routing protocol that produces significantly better performance compared to other well-known scalable routing protocols. Its supremacy is emphatically supported by simulation results as well as mathematical analysis. Imposing some constraints to be followed during division of the network into grid cells has intelligently controlled complexity of our proposed scheme. BinLS is very suitable for highly dense networks of today where mean node density is greater than or equal to 1. Our future work will focus on extension of BinLS in three dimensions.

REFERENCES

[1] Elizabeth M. Royer, A review of current routing protocols on ad hoc mobile wireless networks, IEEE Personal Communications, April 1999

[2] Ian D. Chakeres et. Al, AODV routing protocol – Implementation and Design, Proceedings of WWAN, March 2004

[3] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu and J. Jetcheva, A performance comparison of multi-hop wireless ad hoc network routing protocols, in: Proceedings of 4th Annual Conference on Mobile Computing and Networking (Mobicom 1998), Dallas, TX (October

25-30, 1998).

[4] P. F. Tsuchiya, The Landmark Hierarchy; description and analysis. Technical Report – MTR 87W00152, MITRE Corporation, May 1987

[5] P. F. Tsuchiya, Landmark Hierarchy; architecture, algorithms and issues. Technical Report – MTR 87W00174, MITRE Corporation, May 1987

[6] P. F. Tsuchiya, The Landmark Hierarchy; a new hierarchy for routing in very large networks. In Proceedings of MOBIHOC, Stanford, CA, Aug 2000

[7] G. Pei, M. Gerla and T.W. Chen, Fisheye State Routing In Mobile Ad Hoc Networks. In Proceedings of ICDC Workshop on Wireless Networks and Mobile Computing, pp. 71-78, 2000

[8] G. Pei, M. Gerla, X. Hong, LANMAR: Landmark routing for large scale wireless ad hoc networks with group mobility. In Proceedings of SIGCOMM 1988, Stanford, CA

[9] Seung-Chul M. Woo and Suresh Singh, Scalable routing protocol for ad hoc networks, Wireless Networks, vol. 7 no. 5, pp. 513-529

[10] USCGNavigation Center GPS page, January 2000. http://wwH.navcen.uscg.mil/gps/default.html.

[11] S.M. Das, H. Pucha, Y.C. Hu: Performance of Scalable Location Services for Geographic Ad Hoc Routing, IEEE INFOCOM 2007

[12] C.T. Cheng et. Al. SLALoM: A scalable location management scheme for large mobile ad hoc networks. In Proceedings of Wireless Communications and Networking Conference 2005.

[13] B.C. Seet, Y. Pan, W.J. Hsu and Chiew-Tong Lan, “Multi-Home Region Location Service For Wireless Ad Hoc Networks: An Adaptive Demand-Driven Approach”, Proceedings of IEEE WONS, January 2005

[14] C. Intanagonwiwat, R. Govindan and D. Estrin, Directed diffusion: A scalable and robust communication paradigm for sensor networks, in: Proceedings of 6th Annual Conference on Mobile Computing and Networking (Mobicom 2000), Boston, MA (August 6-11, 2000).

[15] Josh Broch, David A. Maltz, David B. Johnson, Yin-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In Proc. ACM/IEEE MobiCom, pages 85-97, October 1998.

Authors

Anuradha Banerjee is currently working as a lecturer in Kalyani Govt. Engg. College. She completed her B.E. in Computer Science And Technology from Bengal Engineering College (Deemed University), Sibpur, West Bengal, India in the year 2001. She has 12 publications till date, including National and International conferences and journals. Her research area includes Ad Hoc Networks, Neural Networks and Artificial Intelligence.

Dr. Paramartha Dutta is currently working as a Professor in Visva-Bharati University, Santiniketan. He has completed his M.Tech. from Indian Statistical Institute, Calcutta in 1993 and Ph. D. in Computer Science and Engineering from Bengal Engineering And Science University, Sibpur, West Bengal, India in the year 2005. He has 62 publications in National and International conferences and journals. His research interests include Ad Hoc Networks and Image Processing.