5.1 INTRODUCTION

Due to the diversity of applications of ad hoc networks, it is a challenging task to design a single protocol that can operate efficiently across a wide range of operational conditions and network configurations [51, 67]. Considering various MANET applications where mobility, traffic, network size, and node density may vary significantly, different choices and trade-offs have to be made in different situations [47]. Therefore different routing strategies need to be implemented in different phases to achieve good routing performance. However, most of the existing protocols use only one routing strategy for different types of networks. For instance, routing protocols that give good performance for sparse networks may not perform well in dense networks. Similarly routing protocols that are suitable for small networks may not scale well in large networks. In fact, the existing routing protocols suffer from the following problem:

- Unsuitability problem i.e. they cannot be applied to different networks.

The existing location based schemes LAR and DREAM have been described and analyzed in chapter 2. These schemes have following drawback:

- Poor scalability due to flooding of location update and query

Geographic routing protocols also exploit geographic partitions besides the location information of the individual nodes for the routing process. Geographic partition-aided routing can be flat or hierarchical. Protocols based on the geographic partitions generally suffer from the mentioned problem:

- Clogging problem due to cluster-head and hierarchy

The Location Aided Flat Hybrid Routing Protocol (LAFHRP) protocol provides remedy to the problems mentioned above using a hybrid solution of topological and geographic routing to attain the benefits of both the schemes. Poor scalability of the location schemes is
eliminated by avoiding the exploitation of flooding of location update and query messages and also by employing the distributed location service SLS. In spite of using the partitions, it is implemented as a flat routing protocol thereby also eliminates the clogging problem due to cluster-head and hierarchy. LAFHRP also make use of long hop routing to forward the packets to the destination using lesser number of long hops. Using the location information, the control packets are forwarded through limited partitions towards the destination that reduced the control overhead significantly in comparison to network wide flooding. Additionally, the protocol employs a component named Sparse_Forward to perform well in sparse networks besides giving the high performance in dense networks.

5.2 LOCATION AIDED FLAT HYBRID ROUTING PROTOCOL

The LAFHRP is a hybrid routing protocol that provides a hybrid solution of topological and geographic routing. Considering the fact that most of the traffic is directed to nearby nodes in an ad-hoc network, the topological proactive scope is reduced to a small zone around each node in LAFHRP. In a limited zone, all nodes proactively store local routing information; therefore, route requests can be more efficiently performed without querying all the network nodes. Further, it tries to minimize the amount of routing information that is never used. Moreover, the maintenance of routing information also becomes easier [74]. Still, nodes farther away can be reached with the help of geographic information.

5.2.1 ROUTING ZONE

Overlapped k-hop routing zones are defined for each node separately, as described in chapter 4 for BELAHR.

5.2.2 GEOGRAPHIC LOCATION

LAFHRP uses the location information in directional forwarding wherein the sender forwards the data in the relative direction of the destination in hopes of getting it there quickly i.e. communication delay can be minimized using the location information. As shown in Fig. 5.1, node S selects the peripheral node G whose projection SR on line SZ is closest to target Z among S's k-hop neighborhood [97]. A node uses GPS device to obtain its geographic location information. This requirement is quite realistic today because such
devices are inexpensive and can provide reasonable precision. The locations of other nodes can be obtained by using a distributed location service.

![Figure 5.1: 3-hop routing zone at node S with projections](image)

Node S sends data to Z through G because of its proximity with node Z.

Location-update scheme is essential for routing based on geographic routing, which combines the proactive location updates within nodes' local zones and on-demand updates through a distributed location service for far away nodes. Therefore, dissemination of the location update information through the network is a very substantial issue in the protocol. To accomplish this, every mobile node sends location updates comprising its location within k-hop routing zone and to the nodes which lie on the boundary of the partitions using long radio when their distances exceed a certain threshold since their last update (distance-based) [88].

5.2.3 GEOGRAPHIC PARTITION

The protocol also makes efficient usage of geographic partition of the network area. Because of dynamicity of mobile nodes in MANETs, their positions change frequently over time; however, once determined and conveyed to all nodes at initialization time, geographic partition remains unchanged generally.
Geographic partitions divide the entire routing area of the network into regular shapes' subareas, for example, square, circle, hexagon etc. [31]. In LAFHRP, the routing area is divided into equal-sized square-shaped partitions, each with a specific ID as shown in Fig. 5.2. The squared partitions are taken as a matrix and therefore the ID is a pair of x and y component. These square partitions do not overlap; hence avoid the redundancy of overheads. Using a GPS device, each node knows its own geographic position and therefore, its partitions' ID.

![Figure 5.2: Square partitioned routing area](image)

### 5.2.4 FLAT ROUTING PROTOCOL

Despite the use of zones and partitions, the LAFHRP protocol has a flat view over the network as it does not construct the clusters or virtual backbones like in MCEDAR [68]. Therefore, the organizational overhead that occurs in hierarchical protocols due to the assignment of gateways or landmarks for accessing all levels has been avoided. We know that, in hierarchical protocols, nodes of different subnets must send their communication to a higher level subnet that is common to both nodes [28], which may lead to the congestion at specific parts of the network. Being a flat protocol it reduces such possibility of network congestion. Additionally, the non-overlapped partitions in the protocol help in reducing the duplication of the overheads.

### 5.2.5 LONG HOP ROUTING

The protocol employs the long-hop routing (less number of longer hops) as it is more advantageous than short-hop routing (many short hops) in many networks in every aspect.
In fact, long-hop transmission [2, 53] does not inherently cause more interference. In a first order approximation, the control traffic for routing and route maintenance is proportional to the number of nodes in the route. Also, the probability of a route break due to energy depletion and node failure clearly increases with the number of nodes involved, as well as the memory requirements for the routing tables. For a given mobility pattern, longer links live proportionally longer. If channel coding is taken into account, more multi-hop is further penalized due to the necessary encoding and decoding at each hop. Long-hop intuitively should have minimum end-to-end delay due to smaller processing delay for complete route. Forwarding through fewer nodes also has the benefit of fewer chances of collisions and cause low energy expenditure on transmitting. The distance information allows nodes to adjust their transmission powers according to the distance between the nodes and therefore transmission power is used efficiently. This enables to minimize energy required per routing task and to maximize the number of routing tasks that a network can perform.

**Proposals** - Based on the above discussion we are making following proposals:

- Each and every node is capable of determining its current geographic location, geographic partition and time with the help of inbuilt GPS device. To keep the list of neighbors up-to-date each node exchanges location updates with all its neighbors either timer based or distance based. If a node does not receive location update from one of its neighbors, it removes the node from its neighbor list.

- Each mobile node has a long radio with long transmission range.

- Finally, two neighboring nodes \( P \) and \( Q \) have symmetric links, i.e., if node \( P \) can send a packet to node \( Q \), so does node \( Q \) to node \( P \). This condition is easy to be met by adjusting the transmission power of the two nodes.

**5.2.6 DISTRIBUTED SWARM LOCATION SERVICE**

LAFHRP uses the geographical information to limit the flooding of data packets to a small region, rather than to be helpful only during the route discovery phase in LAR. Each node in the network maintains a part of the overall location database. As per swarm location service (SLS) [88], a "location node" is a node that provides location information to other nodes. Although each node has the ability to act as a location node, a node which is rich in
resources like memory, comparatively stable, and having a large broadcasting range is selected by LAFHRP to be location node. In order to facilitate the location service, each node has the following data structures in addition to those needed for the routing algorithm as follows:

- A localized hybrid "location table" is maintained by each node, which contains routing entries for every neighbor within the k-hop routing zone as shown in Table 5.1. Each routing entry contains the IP of neighbor nodes, their location and a timestamp indicating when the entry was added or updated. The location table is amended by adding two more fields that are next hop and total hops. At a minimum, the node stores its own location and the locations of its neighbors in its location table. Assuming that there is enough space available, the node may also store the locations of some other nodes, including those with which it communicates and those whose locations it hears by passively listening to packets being sent on the network. The number of entries stored in the location table is related to the location node’s "goodness" pheromone, described below. However, entries expire from the table after a certain time period, in order to clear a node’s table of possibly outdated information.

<table>
<thead>
<tr>
<th>IP</th>
<th>Location</th>
<th>Next Hop</th>
<th>Total Hops</th>
<th>Timestamp</th>
<th>Partition ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>222.24.15.06</td>
<td>420 10° E</td>
<td>560 40° S</td>
<td>222.24.15.15</td>
<td>3</td>
<td>15:09 PM</td>
</tr>
<tr>
<td>222.24.15.11</td>
<td>550 10° W</td>
<td>340 33° S</td>
<td>222.24.15.31</td>
<td>2</td>
<td>15:15 PM</td>
</tr>
<tr>
<td>222.24.15.20</td>
<td>230 26° E</td>
<td>150 14° N</td>
<td>222.24.15.19</td>
<td>3</td>
<td>15:24 PM</td>
</tr>
<tr>
<td>222.24.15.29</td>
<td>450 30° N</td>
<td>430 20° E</td>
<td>222.24.15.43</td>
<td>1</td>
<td>15:42 PM</td>
</tr>
</tbody>
</table>

- The second data structure that each node has, for supporting the location service is a "scorecard" as shown in Table 5.2. This is a table that contains the IP of a node and a score (pheromone) indicating how "good" the node is at providing location information to other nodes. If the pheromone is a small value, it is not a good location node, and it stores only a small number of other node locations. If the pheromone is a large value, the node stores more node locations.

This pheromone is related to the amount of location information, a node stores. It may be initialized to be proportional to the available size of the node’s location table. When the node answers a source node’s request, pheromone is increased and when
When a node needs the location of another node it sends a location request packet LREQ as shown in Fig. 5.5, to the highest scoring node of its scoreboard. This packet contains the IP and location of the source node, the IP and location of the location node it intends to ask, IP of the destination node and a timestamp.

A location reply packet LRPL, as shown in Fig. 5.6, is sent in response to a location request packet LREQ. The LRPL packet contains the IP and location of the destination node, the IP and location of the location node, the IP and location of the source node and a timestamp.

5.2.7 TRANSMISSION OF PACKETS

In LAFHRP, proactive topological routing operates within the k-hop routing zone. When a node wants to route a packet, it first checks if the destination is within its k-hop routing zone. If destination lies within the range of long radio, the node forwards the packet to the destination directly, otherwise, it forwards the packet to the next hop towards the destination by looking up the location table. For the far away nodes, the node first queries the location of a destination using its own scoreboard and location table of location nodes. After getting the location of the destination, the sender node selects the peripheral node lying on the perimeter of its k-hop routing zone geographically closer to the destination in
comparison to other nodes of the zone. Thereafter, the packet is forwarded to the peripheral node directly and the subsequent packets are unicast only to the target’s nearest peripheral node within their k-hop zone and so on until the packet reaches to the target.

In case of sparse network, if there is no node in the zone then using the long radio the data packet is forwarded to all those nodes of the location table which lies in its own partition and in only two other neighbor partitions having minimum distance from the target node’s partition. Since the traffic would be forwarded only to two nearest neighbor partitions out of eight surrounding partitions so it effectively reduces the traffic and saves the bandwidth a lot. For instance as shown in Fig. 5.2, node A has no node in its zone so it forwards the data packet to node B, node C and node D. Node B is in its own partition i.e. [1, 2], nodes C and D are in the neighbor partitions [1, 3] and [2, 2] respectively nearest to the target’s partition [2, 3].

For the selection of suitable peripheral node, each node assists as per geographic information of its k-hop neighbors stored in the location table. Therefore, to send a packet, a node uses its current location, the locations of each of its neighbor nodes, and the location of the packet’s destination to forward the packet toward its destination.

5.2.7.1 Constrained Diffusion Directional Forwarding Algorithm (CDDFA)

A Constrained Diffusion Directional Forwarding Algorithm (CDDFA), given in Fig. 5.7, searches the target in the location table for the purpose of forwarding the packets. If a node is found within the long radio range of the source node then the packet are directly forwarded to the target, otherwise the packets are forwarded to the next hop towards the target. If the target is not available in the location table then selects the peripheral node of its zone nearest to the target for data forwarding. It may be noted that it actually reduces the scope of data packet forwarding whereas the existing protocols broadcast the data packet to entire ad hoc network. Thus, by not involving the entire network into data forwarding activity, LAFHRP reduces the packet processing considerably. By default, this feature significantly saves the battery power and channel bandwidth.
Algorithm: CDDFA (LOCTB|), LT_SIZE, D_PKT

/* LOCTB is the location table maintained by each node having entries for all neighbors in its zone and some other neighbors in case of free room. LT_SIZE is the maximum entries in the table and D_PKT is the data packet. */

if (LT_Size == 1)  // forward the data packet to only entry in the Location table
    forward (D_PKT, LOCTB [1].IP);
else {
    //search the availability of target in the neighborhood
    i = 1;
    repeat {
        if (LOCTB[i].IP == D_PKT.DA) {
            If(distance(src, LOCTB[i].IP) ≤ long_radio(src))
            /* send D_PKT to the target directly, if target is within the long transmission range of source i.e. long_radio */
            forward (D_PKT, LOCTB [i].IP);
            else //send D_PKT to the next hop towards the target
                forward (D_PKT, LOCTB [i].Next_Hop);
            exit ();
        }
        i++;  
    } until (i > LT_SIZE)

    /* target node is not in the vicinity of source node, so find the peripheral node nearest to the target */
    Peripheral_Node = Target_Nearest_Node();

    if (Peripheral_Node != null)
        forward(D_PKT, LOCTB[Peripheral_Node].IP);
    else
    /* no peripheral node is found in routing zone of source node in case of Sparse Networks */
        Sparse_Forward(D_PKT)
}

Figure 5.7: CDDFA algorithm

5.2.7.2 Target_Nearest_Node Algorithm

Target_Nearest_Node algorithm, as given in Fig. 5.8, selects a peripheral node of the zone which is nearest to the target’s location by comparing the projection of the peripheral nodes
on the direction of destination. Projections are found out using the latitude and longitude of a node.

Algorithm: Target_Nearest_Node()
// finds a zone peripheral node nearest to target's location

i=1;
p_node=1;

// a node with the larger projection on the direction of the target would be nearest to the target
repeat {
  if ( (tot_hop(LOCTB[i]==k) &amp; (tot_hop(LOCTB[i+1]==k)
    (projection(LOCTB[i])< projection(LOCTB[i+1])))
    p_node=i+1;
    i++;
} until (i==LT_SIZE)
return(p_node);

Figure 5.8: Target_Nearest_Node algorithm

Algorithm: Sparse_Forward(D_PKT)
/* find the difference (in degrees) in longitude(λ) and latitude(Φ) of the intermediate node i.e. int_node and the target */
Δλ= λ(int_node) - λ(dest);
ΔΦ= Φ(int_node) - Φ(dest);

/* convert the difference from degrees to distance as per the location of the place */
ΔY = C1 x Δλ;
ΔX = C2 x ΔΦ;
/* find out the two neighbor partitions (i.e. partition1 and partition2) closer to the target by comparing the sign of ΔY and ΔX. P_ID is the partition ID given in the location table. */
If (ΔY < 0)
{ If (ΔX < 0)

  Partition1.x = x of P_ID of int_node;
  Partition1.y = y of P_ID of int_node - 1;
  Partition2.x = x of P_ID of int_node + 1;
  Partition2.y = y of P_ID of int_node;
}
else
{
    Partition1.x = x of P_ID of int_node;
    Partition1.y = y of P_ID of int_node + 1;
    Partition2.x = x of P_ID of int_node + 1;
    Partition2.y = y of P_ID of int_node;
}

else // examine the case of ΔY being greater than 0
{
    If (ΔX < 0)
    {
Partition1.x = x of P_ID of int_node;
    Partition1.y = y of P_ID of int_node - 1;
    Partition2.x = x of P_ID of int_node - 1;
    Partition2.y = y of P_ID of int_node;
}
else
{
Partition1.x = x of P_ID of int_node;
Partition1.y = y of P_ID of int_node + 1;
Partition2.x = x of P_ID of int_node - 1;
Partition2.y = y of P_ID of int_node;
}
}

/* forward the packet to all neighbor nodes within the same partition and two other adjacent partitions closer to the target */
i = 1;

repeat

If((P_ID.x==int_node.x) & (P_ID.y==int_node.y)) OR (P_ID.x==partition1.x) & (P_ID.y==partition1.y)
    OR (P_ID.x==partition2.x) & (P_ID.y==partition2.y)

forward (D_PKT, LOCTB [i].IP);

i++;  
until (i > LT_SIZE )

Figure 5.9: Sparse_Forward algorithm
5.2.7.3 Sparse Forward Algorithm

In case of sparse network, if a node does not have immediate neighbors within its zone then Sparse Forward algorithm, as given in Fig. 5.9, forwards the packet selectively to those nodes, that are present in the location table and belong to either its own partition or to the two partitions out of eight surrounding partitions nearest to the target's partition.

Sparse Forward algorithm ensures the shortest route and rules out the possibility of packets drop out in case of low density of nodes in the network.

5.3 ANALYSIS OF LAFHRP ALGORITHM

LAFHRP provides a number of benefits because of its hybrid nature, location advantage of nodes, long radio facility in nodes, constrained directional forwarding of packets towards target and its flat nature even after using the geographic partition.

**Suitable for dense networks as well as sparse networks** — Its hybrid nature makes it suitable for different networks in terms of reduced overheads and efficient utilization of bandwidth and energy resource. It also performs well in sparse networks with the help of the Sparse Forward component.

**Reduced overheads**— Being a flat hybrid routing protocol it reduces the overheads otherwise required for the organization and maintenance of the clusters. Since, swarm intelligence location tracking is used to determine the location of the target, data packet forwarding is achieved with minimum overheads due to the reduced scope of forwarding in the targets’ direction only. Also the protocol rely on constrained diffusion for location update and query rather than exploiting flooding which ensures the reduction in overheads.

**Independent of single point failure**— Being a flat routing protocol, it reduces single-point of failure as there are no cluster-heads for partitions.

**Minimum cost route and minimum end-to-end delay**— Due to long hop routing, LAFHRP uses the directional path through the partitions thus providing a minimum cost routes for
data forwarding towards the destinations. The protocol is also effective in minimizing the end-to-end delay and also the bandwidth as a chaining effect.

*Updated routing information*- Collaboration between nodes also help in maintaining routing information much longer and also updated. This may potentially eliminate the need for flooding, since the nodes know exactly where to look for a destination every time.

### 5.4 CONCLUDING OBSERVATIONS

The described protocol is a flat routing protocol as it does not construct clusters or virtual backbones. It doesn't exploit flooding for location update and query. The protocol is well suitable to sparse networks besides the dense networks. It is efficient and scalable because it combines the merits of both topological and geographic routing. Location service augmented with MANET protocols improves the scalability by minimizing the routing overheads which in turn minimizes wireless bandwidth consumption. Also, end-to-end delay is minimized due to long hop routing. Limitation of the protocol is its inefficiency in case of fast moving mobile ad hoc networks.