Technical Report

UCAM-CL-TR-626
ISSN 1476-2986

Number 626

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Landmark Guided Forwarding:
A hybrid approach for Ad Hoc routing

Meng How Lim, Adam Greenhalgh,
Julian Chesterfield, Jon Crowcroft

March 2005

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

Technical reports published by the University of Cambridge
Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/TechReports/

ISSN 1476-2986

Landmark Guided Forwarding: A hybrid approach for Ad Hoc routing

MH.Lim[§*] **A.Greenhalgh**[‡†] **J.Chesterfield**[§] **J.Crowcroft**[§]

[§]Computer Laboratory,University of Cambridge,UK

{*meng.how − lim, j.chesterfield, jon.crowcroft*}@*cl.cam.ac.uk*

[‡]Dept. of Computer Science,University College London,UK

a.greenhalgh@cs.ucl.ac.uk

Abstract

Wireless Ad Hoc network routing presents some extremely challenging research problems, trying to optimize parameters such as energy conservation vs connectivity and global optimization vs routing overhead scalability. In this paper we focus on the problems of maintaining network connectivity in the presence of node mobility whilst providing globally efficient and robust routing. The common approach among existing wireless Ad Hoc routing solutions is to establish a global optimal path between a source and a destination. We argue that establishing a globally optimal path is both unreliable and unsustainable as the network diameter, traffic volume, number of nodes all increase in the presence of moderate node mobility. To address this we propose *Landmark Guided Forwarding* (LGF), a protocol that provides a hybrid solution of topological and geographical routing algorithms. We demonstrate that LGF is adaptive to unstable connectivity and scalable to large networks. Our results indicate therefore that Landmark Guided Forwarding converges much faster, scales better and adapts well within a dynamic wireless Ad Hoc environment in comparison to existing solutions.

1 Introduction

Ad Hoc networking is a topic of widespread interest amongst the network and systems research communities of late due to the novel challenges associated with providing truly distributed and decentralized communication architectures. Building Ad Hoc networks over wireless links introduces even more complexity due to the irregular and unpredictable nature of the wireless medium. Regardless of these considerations, however Ad Hoc wireless systems are growing rapidly, fueled in large part by the proliferation of cheap, Local

*MH.Lim's PhD work is funded by the Cambridge-MIT Institute

†A.Greenhalgh's post doctorial research is funded by the Cambridge-MIT Institute

Area wireless technologies conforming to the IEEE 802.11 family of protocols.

The primary objective of wireless Ad Hoc networks is to enable a set of highly cooperative wireless nodes to establish communication quickly without any infrastructure and exchange data amongst themselves. In addition to sending and receiving packets, each node also acts as a relay for packets traveling across the network from a source which may not be able to directly access the destination node, for example as a result of signal power limitations, or due to the well known 'hidden node' problem.

Unlike mobile hosts in an infrastructure based mobile network, such as an office or home setting with a dedicated wireless base station, nodes in a Mobile Ad Hoc Network (MANET) must collectively manage the communication infrastructure in a cooperative fashion amongst themselves. MANETs rely on the common sharing of resources to achieve a collective goal.

There are a variety of issues and solutions surrounding the development of social and economic models to provide incentives for cooperative network architecture formation, such as is required in the Ad Hoc scenario, which we do not address in this paper. Our work concerns the mechanics of building an Ad Hoc routing infrastructure, and as such builds upon a substantial body of research. However, our work differs from existing approaches in a number of respects.

Whilst many routing protocols utilize either topologically driven route optimization, or geographically driven route optimization, we maintain that a more efficient approach is to leverage benefits from each, creating a hybrid approach towards routing, optimized around various local and global parameters. In this paper we present *Landmark Guided Forwarding* (LGF), a novel approach towards Ad Hoc network routing that achieves the following:

- Increased *global resilience to incorrect device positioning* information
- *Lower average routing state* maintenance across the node set
- *Lower network routing overhead*

Unlike topological routing protocols such as [3, 4, 13], *Landmark Guided Forwarding* requires that every node only maintains a small amount of topological information about its' neighbors within a localized area. Routing is achieved by using locally optimized algorithms, requiring lower network overhead. If the packet destination resides within the local area, it is routed using the shortest path algorithm. Otherwise, when the destination resides outside the local scope, it is routed towards a geographically determined optimal Landmark node. Unlike position based forwarding schemes such as [1], LGF does not rely upon the establishment of a globally optimal path across Landmark nodes, but leverages the local topological routing information available, thereby increasing the resilience to inconsistent device position information and lowering the overall system vulnerability to position errors [7].

In the remainder of this paper we examine similar related work in the field and how it compares to LGF and the assumptions we have made while developing LGF. We follow on to describe LGF in detail before describing how we simulated LGF in different scenarios and the results we obtained. Finally we summaries the results and conclude before suggesting some possible future extensions to this work.

2 Related work

Many wireless Ad Hoc wireless routing protocols have been proposed in recent years. An early survey paper [12] categorized these protocols as table driven or source driven. In general, table driven protocols pro-actively gather topological routing information while source driven protocols reactively discover a route or routes to the destination as requested by the source. Pro-active routing protocols such as DSDV, Destination Sequenced Distance Vector [3], pro-actively exchange routing information between neighboring nodes. The associated routing state and the network traffic overheads is $O(n)$, where n is the number of nodes in the network, which does not scale well in large networks. Reactive routing protocols such as DSR [4], Dynamic Source Routing and AODV [13], Ad Hoc on Demand Distance Vector, use flooding techniques to discover new routes and repair existing routes. As the amount of traffic in the network increases or the diameter of the network increases, the cost of flooding increases. With reactive routing protocols the routing performance degrading under moderate mobility conditions [11][16].

An alternative approach to Ad Hoc routing is to take advantage of the physical location of nodes in the network and to do position based forwarding. An assumption made by protocols that take this approach is that every node knows its own geographical position. By limiting the exchange of positional information to be only between adjacent nodes, the state and network overheads are reduced to $O(u)$, where u is number of adjacent nodes. GPSR, Greedy Perimeter Stateless Routing [1], is a position based routing protocol that in general uses the geographically closest node to the destination as the next hop for the packet to be forwarded . This technique can result in local maximum in its proximity to the destination where greedy forwarding would prevent the packet from advancing towards the destination. To address situation like this, GPSR uses a perimeter forwarding scheme that uses the well known right hand rule on its planarised graphs. Although GPSR scales well and adapts to random topologies, its perimeter forwarding method can direct the packet along a suboptimal route in a large network where the shortest geodesic path between the source and destination is not well connected [9]. A recent research article suggests that position based forwarding protocols are vulnerable to position errors [7]. The results show an increase in packet dropping and routing loops that is correlated with the magnitude of inconsistencies in the position of a device. For example if inaccuracy in a device's position is 20 percent of its radio range or less, then up to 54 percent of packets maybe dropped.

LGF is similar in some of its features to existing routing protocols, such as ZRP [6] and Terminodes [9]. In common with these two protocols LGF uses a hierarchical framework that employs two different routing schemes. Each node pro-actively maintains connectivities with other nodes within its neighborhood, a packet is routed using the shortest path algorithm when the destination is within this neighborhood. In contrast, a packet destined for outside the local neighboring is routed using a more scalable routing protocol. ZRP uses reactive routing to determine the optimal path to the destination by using flooding where as Terminodes uses position based forwarding to route a packet towards the geographical location of the destination. Unlike ZRP that maintains the global optimal route between a source and destination, LGF progressively explores the area through a geographic depth first search. In contrast to ZRP, the Terminodes routing protocol uses greedy forwarding to forward packets, but it requires some static nodes to establish stable paths when greedy forwarding is not applicable.

Position based forwarding protocols in general use greedy forwarding schemes to propagate packets from a source to a destination. Greedy forwarding provides no guarantees about successful packet delivery even if there exists a path between source and destination. Message delivery guaranteed was first introduced by the FACE routing algorithms [2]. The FACE-2 algorithm traverses the whole face of planar graph using the right hand rule until the destination is found and the packet is routed along this path. GPSR [1] combines greedy forwarding with the concepts of FACE routing.

In the a paper title "A Distance Routing Effect Algorithm for Mobility (DREAM)"[15], it is observed that "*distance effects*, uses the fact that the greater the distance separating two nodes, the slower they appear to be moving with respect to each other". The DREAM protocol uses this observation and the actual node movement to control the speed of the routing and position updates for nodes by relating them to their apparent "speed". A limitation of this technique is that every node floods all other nodes with its position and velocity. For the exchange of routing information, a device uses the last known position and velocity of the a packet's intended destination to calculate a circular area where the destination is likely to be found before. DREAM uses restricted flooding techniques to flood packets for the destination within this circular area. Each device using DREAM maintains global state but allows nodes further away to update less frequently while in LGF we use mobility sensitive advertisements with restricted propagation of these advertisements, along with local optimal routing.

LGF takes a similar approach to the Grid Location Service [10], GLS, which uses a two hops distance vector scheme to alleviate holes in the topology whilst applying greedy forwarding techniques. We acknowledge that this approach is similar to our dead end detection scheme which is an intuitive approach that uses a depth first search with soft-state and source path state. As the main focus of GLS was not on routing, it does not propose a complete routing solution to resolve cases where greedy forwarding is not sufficient.

3 Assumptions

We make a number of assumptions in LGF. Firstly, we assume that every node knows its own geographic position. This is not an unreasonable assumption since it is feasible to gather position information from GPS or another positioning system. Since LGF does not require high precision, short range distance measurements from Bluetooth devices or via IEEE 802.11 based ranging systems such as the Intel Place Lab system [8]. Secondly we assume a distributed location service like the Grid Location Service[10] is available for use by LGF. Lastly we assume circular radio propagation area.

4 Protocol description

The theoretical analysis of IEEE 802.11 wireless Ad Hoc networks [11] states when the number of nodes randomly distributed throughout a unit kilometer square is greater than $(6/r^2)\ln(6/r^2)$, where r is the radio propagation radius, then dead ends are extremely unlikely to occur. This analysis also demonstrates that handshaking and interference in the IEEE 802.11 protocol can significantly degrade the performance of multi-hop routing. For example if 10 nodes are arranged in a chain, the chain of nodes is only able to achieve 1/7 of its maximum throughput. It was also show by Li et. al [10] using simulation results that the fraction of undelivered packets is 0.02 for a node density of 75 nodes per km^2, with nodes moving randomly at 10 m/s. This clearly shows the correlation between density and random network connectivity in wireless Ad Hoc networks. Overall these findings need to be considered when designing wireless Ad Hoc routing protocols that use the 802.11 communications standard.

Ad Hoc networks rely on nodes in the network to relay packets between a source and a destination on behalf of their peers. As a packet flows between the source and destination, LGF calculates the locally optimal path to the destination and applies the shortest path to the destination if it is within range of local area. In cases where destination is not within the local area, it applies local optimal routing to the node that is the geographically closest to the destination. The protocol is progressive. Once the packet is forwarded, it will uncover a new set of neighbors and local optimal route towards the destination. Using this technique not only effectively unlocks the scalability constraints associated with global optimal routing as used by existing MANET protocols but also allows routing to be more adaptive to the ever changing MANET topology. As the approach taken by LGF only requires advertisement of topological and geographical information to a node's neighbors that are within a few hops (currently 3 hops), it localizes state dissemination and reduces the overall load on the network. In essence, these properties allow LGF MANETs to be extended to a wider environment.

In this section, we present the various algorithms that form Landmark Guided Forwarding. The protocol consists of seven components, namely: restrictive hybrid route advertisement, adaptive route advertisement, link failure maintenance, next hop selection algorithm, path exploration, dead-end detection and loop avoidance. We describe each of these in turn in the later sections.

4.1 Restrictive hybrid route advertisement

In order to retain a balance between timeliness of routing decisions and the overhead of route advertisement, we proposes a pro-active routing scheme based on a localized hybrid routing table. Using this approach, information about a node's geographical position and local topology is disseminated to a limited topological area. We define each node's neighbors to be within a topological area defined by the perimeter P. For each neighbor node j, node i maintains its position, x_j, y_j, z_j, and the distance between node i and node j, d_{ij}. Each node i maintains this information and additional information as a routing entry RE_{ij}, in the routing table RT_i. For each neighbor j a node i may hold multiple routing entries, RE_{ij}, a sequence number s_{ij} is associated with each entry to ensure timeliness. A routing entries RE_{ij} is given below:

$$RE_{ij} = \{j,\ NextHop_{ij},\ HopCount_{ij},\{x_j, y_j, z_j\},\{\dot{x_j},\ \dot{y_j},\ \dot{z_j}\ \},\ Seq.Num._{ij}\}$$

Where j is the destination and is a globally unique node identifier of all nodes within P, and the *Next Hop* is the identifier of adjacent node that a packet should be forwarded in order to reach *destination* which is $HopCount$ hops away. The position and velocity of the destination j, are $\{x_j, y_j, z_j\}$,$\{\dot{x_j},\ \dot{y_j},\ \dot{z_j}\ \}$ respectively. These attributes are used by the forwarding algorithm to resolve a local optimal path when destination address dst_p of a packet p is not in RT_i, $\forall j, dst_p \neq j$.

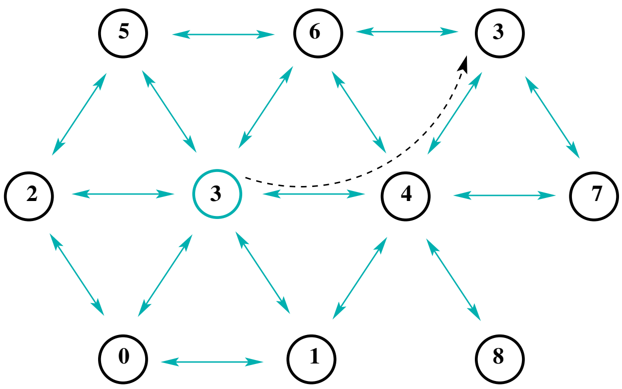

Figure 1: Mobility scenario in an Ad Hoc network

In order to explain the restricted hybrid routing advertisement process we use an example. Figure 1 shows a small Ad hoc network scenario where node 3 moves from its central position to a new position in the top right of the network, all other nodes remain stationary. We demonstrate the scheme by comparing the routing tables and the topological view of the network from node 5's point of view.

In this example routing information does not propagate more the 2 hops from the source. If we look at Node 5's routing table, table 1, and its topological view of the network, as illustrated in figure 2.

If we re-examine node 5's routing table, table 2, and its topological view of the network, figure 3, after the movement of node 3. Node 1 is now no longer routable using local optimal routing, nodes 3 and 4 are now only routable via node 6 and node 0 is only routable via node 2.

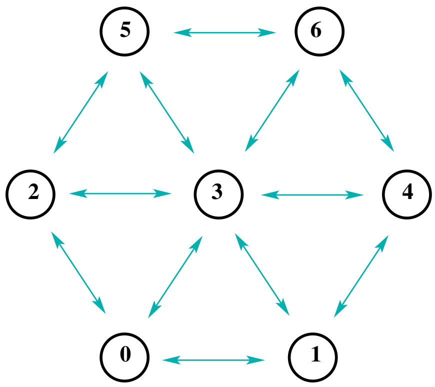

Figure 2: Node 5's topological view of the network before node 3 moves.

Table 1: Node 5's routing table before node 3 moves.

Dst	Next Hop	Metric	x	y	z
0	2	2	300.00	2.00	0.00
1	3	2	450.00	2.00	0.00
2	2	1	225.00	132.00	0.00
3	3	1	375.00	132.00	0.00
4	3	2	525.00	132.00	0.00
5	5	0	300.00	262.00	0.00
6	6	1	450.00	262.00	0.00

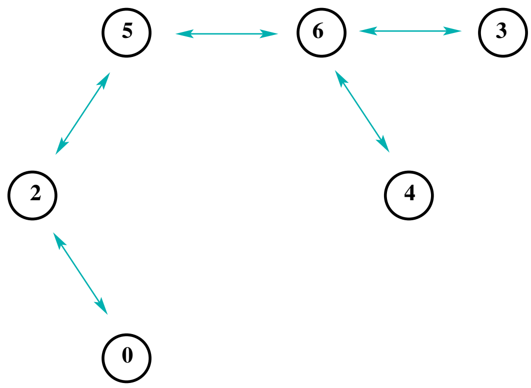

Figure 3: Topological view of node 5 after node 3 move away

Table 2: Routing table of node 5 after node 3 move Away

Dst	Next Hop	Metric	x	y	z
0	2	2	300.00	2.00	0.00
2	2	1	225.00	132.00	0.00
3	6	2	600.00	262.00	0.00
4	6	2	525.00	132.00	0.00
5	5	0	300.00	262.00	0.00
6	6	1	450.00	262.00	0.00

Node 3's movement causes the routing algorithm to make the following adjustments to routing table of node 5, i.e. table 1 is transformed to table 2.

- Remove entries for destinations which have a hop count greater than 2.
- Update of location information
- Update of next hop and metric information

Specifically it can be observed that entry for node 1 has been removed from routing table of node 5 in table 2. The position of node 3 has been updated and the next hop and metric of nodes 3 and 4 have been updated accordingly.

Routing updates are sent out as part of the route advertisement procedure which we describe later. These routing updates are processed upon reception by the *RouteUpdate* procedure. The procedure accepts only advertised routes that report link failures ($HopCount = \infty$) or are within a node's topological perimeter ($HopCount \leq P$). This reduces the state propagation and enables this protocol to scale to large networks. The *RouteUpdate* is as follows.

---

procedure *RouteUpdate* ***begin***

if *broadcast route advertisement is received*
 for *all neighbors of sender in route advertisement*
  if *neighbor's HopCount is within parameter or infinity*
   invoke *update routing table*
  end if
 end for
end if

end procedure

4.2 Adaptive route advertisement

We define $Max\_Distant$ as the distant between a node and its furthest adjacent node. In figure 4, the $Max\_Distant$ is the distant between node 1 and 4.

The function *GetDistance* is used to calculate an approximate distant between a node and its adjacent neighbor using Pythagoras's theorem. *GetDistance* uses an estimate of the current location of an adjacent node which is derived from a combination of the last advertised position and the node's velocity, together with $\triangle t$, the time difference between current time and the time when the last advertisement was received from the respective adjacent node.

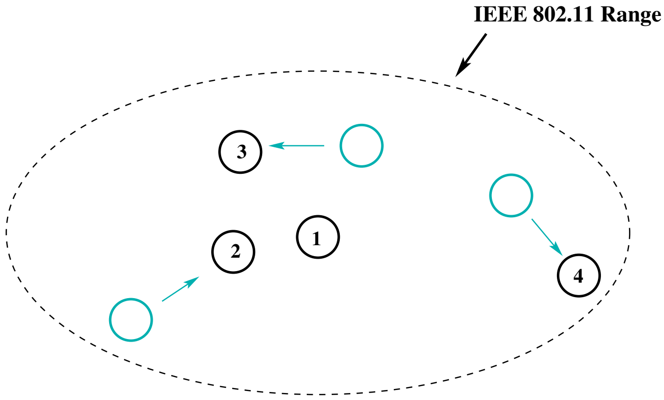

Figure 4: Relative displacement of all one hop neighbors

function *GetDistance(node, neighbor)* ***begin***

$XPos \leftarrow (neighbor.x + (\triangle t * neighbor.dx))$
$Xdist \leftarrow (node.x - XPos)^2$
$YPos \leftarrow (neighbor.y + (\triangle t * neighbor.dy))$
$Ydist \leftarrow (node.Y - YPos)^2$
$dist \leftarrow \sqrt{Xdist + Ydist}$
return *dist*

end function

The *GetDistance* function is used by *GetMaxDistance* function that iterates through all the one hop neighbors in the routing table to find the node which is furthest away.

---

function *GetMaxDistance(node, routing_table)* ***begin***

Max_Distant ← *0*
Distant ← *0*
for *all routing entries in the routing_table*
  if *neighbor is one hop away*
    $Distant \leftarrow GetDistance(node, routing\_entry)$
    $Max\_Distant \leftarrow \max(Max\_Distant, Distant)$
  end if
end for
return $Max\_Distant$

end function

The *max_Distant* is used to adapt the expiry time of a routing using the *Mapping* function. The *Mapping* function is graphically represented in figure 5. As shown in below

RouteAdvertisment procedure, this expiry is used to schedule the next route advertisement. The mapping function is a function that maps the expiry time for a routing entry to the distance the furthest node is away. The function uses the 802.11 radio propagation characteristics to bound the maximum and minimum expiry values.

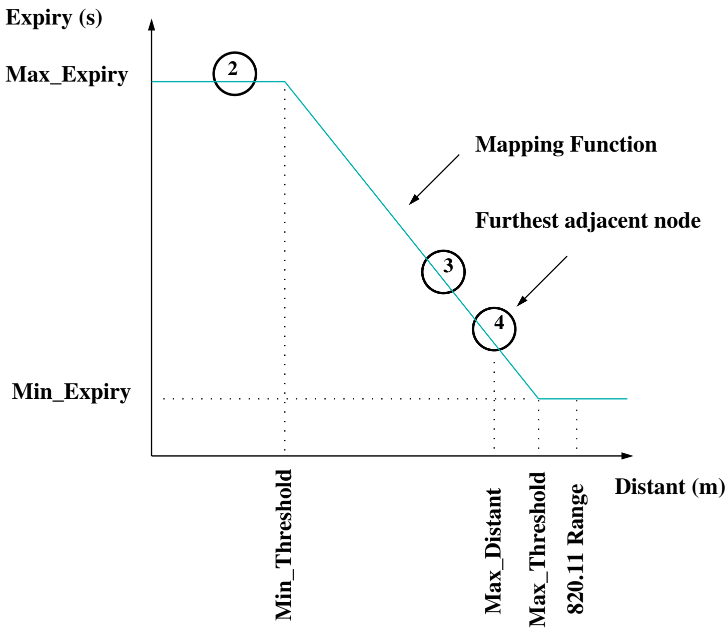

Figure 5: Mapping of furthest adjacent node to the expiry time of routing entries.

---

procedure *RouteAdvertisement* ***begin***

acquire node's geographical information
$Max\_Distance \leftarrow GetMaxDistance(node, routing\_table)$
$expiry \leftarrow Mapping(Max\_Distance)$
insert expiry and geographical information into route_entry
invoke *CreateRouteAdvertisement*
invoke *scheduleNextRouteAdvertisement(expiry)*
invoke *broadcast route advertisement*

end procedure

The *Mapping* algorithm defines the *expiry* time to be inversely proportional to $Max\_Distance$ when $Min\_Threshold < Max\_Distance < Max\_Threshold$. Within this range, the expiry time is determined by $((Radiorange * tunningfactor)/Max\_Distance)$. In order to avoid excessively short lived routing advertisements when link failure is imminent, the minimum expiry time is applied when the maximum distance is greater than $Max\_Threshold$. Conversely, the maximum expiry time is applied to reduce the frequency of updates for near adjacent node when $Max\_Distance < Min\_Threshold$.

---

procedure *Mapping(Max_Distance)* ***begin***

$expiry \leftarrow 0$
if$(Max\_Distance < Min\_Threshold)$
$expiry \leftarrow Max\_Expiry$
else if$(Max\_Distance > Max\_Threshold)$
$expiry \leftarrow Min\_Expiry$
else
$expiry \leftarrow (radiorange * tunningfactor)/Max\_Distance$
end if
return *expiry*

end procedure

4.3 Link failure maintenance

When a node moves out of range of it's neighbors, established links are likely to break. Typically a broken link may be detected either by the link layer protocol timing out a connection, or it may be inferred at a higher level through the loss of a periodic broadcast signal which is expected within a predefined time. In our protocol, a node represents a broken link with ∞.

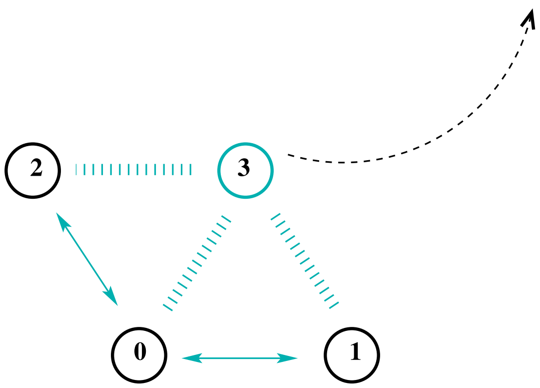

Figure 6: State propagation and maintenance

Figure 6 illustrates a mobility scenario in which node 3 moves out of range of nodes 0, 1 and 2. Node 1 is initially a neighbor of node 3, and records a route to node 3 with a metric of 1 as shown in table 3. As node 3 moves out of range, the node detects the loss of a link, and updates it's table accordingly. Table 4 illustrates the change in routing metrics; the routes to both node 3 and node 2 which originally travelled via 3 are set to ∞. Node 1 subsequently broadcasts these routing entries to all it's single hop neighbors. Once the routing state has been synchronized in this manner, the node performs a periodic state maintenance process, removing or replacing the entries with ∞ metrics with cheaper routes.

Table 3: Routing table of node 1 before link broken

Dst	Next Hop	Metric	x	y	z
0	0	1	300.00	2.00	0.00
1	1	0	450.00	2.00	0.00
2	3	2	225.00	132.00	0.00
3	3	1	375.00	132.00	0.00

Table 4: Routing table of node 1 after link broken

Dst	Next Hop	Metric	x	y	z
0	0	1	300.00	2.00	0.00
1	1	0	450.00	2.00	0.00
2	3	∞	225.00	132.00	0.00
3	3	∞	375.00	132.00	0.00

Table 5: Routing table of node 1 after state maintenance

Dst	Next Hop	Metric	x	y	z
0	0	1	300.00	2.00	0.00
1	1	0	450.00	2.00	0.00
2	0	2	225.00	132.00	0.00

4.4 Next hop selection algorithm

Our approach is to take advantage of the geographical position of those nodes that are within each node's topological scope as a basis for the forwarding algorithm. Each node i maintains the distance d_{ij} and position x_j, y_j, z_j for every other node j that is within its topological scope. The next hop is selected using the shortest path algorithm to each packet's destination d, if it is found in the set J with $j \in J$. Otherwise, the next hop is determined by Landmark Guided Forwarding. The term Landmark has been widely used to describe a physical point of reference for an Internet coordinate system [17]. In this paper, Landmark is a temporary reference node amongst the set J, that acts as a virtual destination to assist in the routing of a packet towards its final destination. The exploration algorithm is progressive, as soon as the packet moves to the next hop, a new Landmark node amongst the new set of neighbors would be determined and the packet would progress in the same manner until it arrives at a node with a topological path to the destination. However, in the case where no valid Landmark node is available for forward advancement, the path exploration algorithm rolls back and seeks an alternate path from the previous hop.

Figure 7 shows a subgraph that demonstrates our forwarding algorithm where the topological scope is limited to 2 hops. If we consider the packet arrives at node 0, can be forwarded to the destination via either node 1, 3 or 5 and the destination for the packet is node 4, by applying the shortest path algorithm to the destination, the next hop is found to be node 3. In the case where a packet's destination is not within the coverage of the

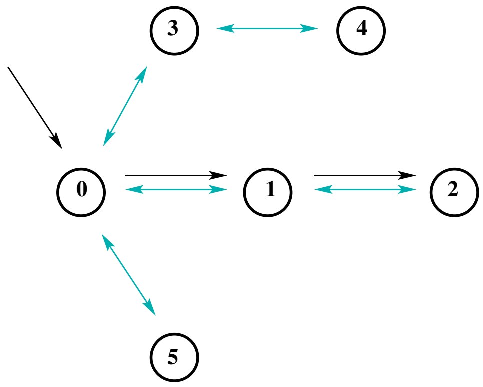

Figure 7: Next hop selection

topological scope, the next hop is chosen by the shortest path algorithm to a landmark node V. Where node V is geographically closer to the destination D and topologically further away from i. For this example in fig 7, the next hop is node 3 if node 4 is found to be closer to the destination than node 2.

As shown below, the *forwarding* function returns a route entry containing the packet's next hop. The function returns an entry to the destination if the routing table has an entry to the destination. If the destination is not within the table, an entry containing the next hop towards a Landmark node is returned. Where a Landmark node is not available, the function returns an entry for packet to retreat to its previous hop.

---

function *Forwarding(packet)* ***begin***

$route\_entry \leftarrow NULL$
$route\_entry \leftarrow found\_in\_routetable(packet)$

if$(route\_entry \neq NULL)$
return *entry*
end if

$route\_entry \leftarrow Get\_Landmark\_Entry(packet)$

if$(route\_Entry \neq NULL)$
return *entry*
end if

$route\_entry \leftarrow Get\_Reverse\_Entry(packet)$
return *entry*

end function

4.5 Path exploration

In general, geodesic proximity to the destination does not assure a shorter topological path to the destination. Simply forwarding a packet towards its' destination position without maintaining any forwarding path history does not provide any facility for preventing the packet being trapped by a localized loop or dropped due to a routing dead-end and subsequently backtracking. The approach adopted in our algorithm is to include a source path in the packet header and to also maintain soft forwarding state amongst all nodes traversed by a packet. By maintaining a source path in the packet header it provides a trail of forwarding nodes such that in the event a dead-end is encountered, the packet can be back-tracked until it reaches a node with an alternative path to the destination. The purpose of maintaining soft-state within the network is to isolate and explore the network systematically. A node temporarily marks a link with the tuple {*Packet_Sequence_Number, Next_Hop, Soft_State_Expiry*}, once it has forwarded a packet along that link. It is assumed that packets travel much faster than nodes move, and therefore it is feasible to do path exploration.

---

function *GetLandmarkEntry(packet)* ***begin***

route_entry ← *NULL*
min_distant ← ∞

for *all routing entries in the routing table*
 if *(found_In_Softstate(packet, entry.next hop)*
  continue
 end if

 if *(found_In_Sourcepath(packet.source_path, entry.dst)*
  continue
 end if

 if *(entry.metric == topology_range)*
  $distant = \sqrt{(X_{packet} - X_{entry})^2 + (Y_{packet} - Y_{entry})^2}$
  if *(min_distant > distant)*
  min_distant = distant
  route_entry = entry
  end if
 end if
end For

return *route_entry*

end function

As shown in the *Get_Landmark_Entry* function above, verifying that a forwarding path is loop-free is done by ensuring the destination of the route entry *entry.dst* does not match any node identifier of the packet's source path *packet.source_path*. In addition, visited links are isolated by soft-state to allow systematic path exploration. In order to detect dead-ends earlier in the forwarding path, the function only accepts Landmark nodes that are within the *topology_range*. In a situation where multiple Landmark nodes are available, the function chooses the node that is closest to the destination.

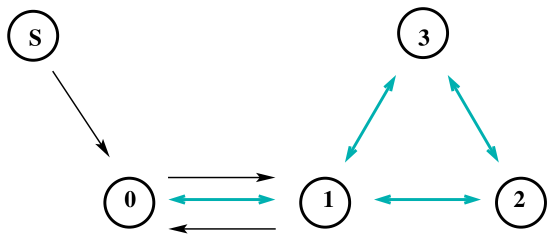

Figure 8: Dead end detection and roll back

Figure 8 shows a subgraph that demonstrates how a dead end can be detected while a packet systematically explores a path to the destination. In this scenario, a packet from node S arrived at node 0. Assume the packet's destination is geographically remote and outside the geographical scope of node 0 and in addition, the destination is geographically closer to node 2 than node 3, the topological scope being 2 hops. We denote SP as a set of nodes in the source path. At node 0, where $SP = \{S\}$, we determine the next Landmark node, according to our next hop selection algorithm, as node 2. The next hop node chosen to forward the packet towards node 2 is node 1 based on the shortest path algorithm.

When the packet arrives at node 1, $SP = \{S, 0\}$. It becomes apparent that the only node that is 2 hops away from node 1 is S. Since S is found in the source path SP, the path exploration detects that the packet is moving towards a dead-end and retracts the packet back to node 0. In this example, node 0 had established soft-state when the packet was forwarded from node 0 to node 1 and likewise node S had established soft-state when the packet was forwarded from node S to node 0. Retracting back to node 0, the packet's source path SP is shortened to $\{S\}$. At this point, the path exploration is aware that the link between node 0 and 1 has already been visited. Since there is no forwarding path available, the packet is pulled back to node S. With no other link available at node S, the path exploration has exhausted all searches and drops the packet.

Figure 9 shows a subgraph that demonstrates how a loop is avoided while a packet explores a path to its destination. In this scenario the topological scope is 2 hops and the source node is S. The destination D is not directly connected to any node in the subgraph. Based on our next hop selection algorithm, the packet at node S identifies node 2 as it Landmark node. Following the shortest path algorithm to node 2, the packet is directed towards node 1. Subsequently, the packet is forwarded to node 2 with Landmark node 4.

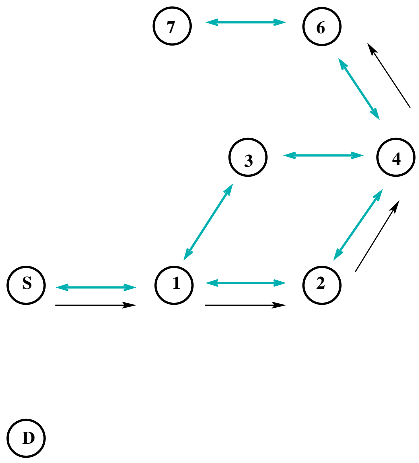

Figure 9: Loop avoidance

The same process is repeated when the packet moves from node 2 to 4 with node 3 as its respective Landmark node. When the packet arrived at node 4, it found $SP = \{S, 1, 2\}$ with both node 1 and 7 within its' topological range, i.e. within 2 hops of node 4. With node 1 in its' source path, the algorithm provides only one option of forwarding towards node 6 with node 7 as the Landmark node. This effectively avoids the creation of a loop between $1 \rightarrow 2 \rightarrow 4 \rightarrow 3$.

5 Simulation scenario

The simulations have been carried out using the NS2 simulator [5], with each simulation lasting for 900 seconds. Each node uses the IEEE 802.11 MAC and physical models with the radius of the radio range being 250 meters. The simulation uses the random way point model to model node mobility. In all simulation scenarios, each node select a random destination and moves at a speed uniformly distributed between 0 and maximum speed. Upon reaching the destination, the node selects the next random destination and moves on. The traffic model uses constant bit rate UDP traffic flows, with 512 byte payloads. The start time for the different flows is uniformly distributed between 0 and 180 seconds with each of the 30 traffic sources at the rate of 2 packets per second. In common with other protocol evaluations, [1][3][13][4], we run several mobility patterns with different pause times at a constant speed. We use 5 different sets of mobility patterns generated with different pause times of 0,30,60,120,600 and 900 seconds and a maximum velocity of 15 m/s. We use two different geographic areas, firstly 50 nodes in an area of 1500x300 m^2 and secondly 100 nodes in an area of 1500x 500 m^2. For mobility scenarios that have a pause time of 0 seconds, i.e. the nodes constantly move, we repeat the simulations with different maximum velocities of 1, 2.5,5,7.5,10,12,5,15 m/s. We compare LGF with DSDV, AODV and GPSR using the different simulation scenarios we have just described and we compare the adaptability, performance and overheads of LGF with other MANET routing protocols. Each of the different Ad hoc routing protocols has some settings specific to it, we detail these in the tables below.

Table 6: GPSR specific parameters

Parameter	*Value*
Beaconing interval	3 s
Random variation of beaconing interval	0.5 %
Beacon expiration interval	13.5 s
Promiscuous mode	enable
Removal of neighbor from neighbor list when link broken	enable
Perimeter mode	enable

Table 7: DSDV specific parameters

Parameter	*Value*
Initial weight settling time	6 s
Periodic update interval	15 s
Number of missed periodic updates before declaring link broken	3
Settling time weight	7/8

Table 8: AODV specific parameters

Parameter	*Value*
Lifetime of a route reply message	10 s
Time for which a route is considered active	10 s
Time before route request message is retired	6 s
Time which the broadcast id for a forwarded route request is kept	6 s
Number of route request retries	3
Maximum route request timeout	10 s
Local repair wait time	0.15 s

Table 9: LGF specific parameters

Parameter	*Value*
Tuning factor	1.5
Max expiry	15 s
Min expiry	1.2 s
Topological scope	3

6 Results

The results are divided into three subsections: performance with varying pause time, performance with varying velocity and path length.

6.1 Performance with varying pause time

Figure 10 evaluates the reliability of packet delivery of the different routing protocols; LGF, GPSR, AODV and DSDV. In general, DSDV, GPSR and AODV perform better as the pause time used in the random way model increases. In contrast, LGF is more robust at higher mobility, the results indicate that its packet delivery ratio is relatively poor when compare to other protocol at low mobility. This is largely due to the way in which LGF handles link failures, GPSR, AODV and DSDV optimize the handling of link failure for stale connectivity, in contrast in LGF, we drop packets as soon as we see the link fail. This is design decision to decrease the average packet delay at the cost of reducing the delivery ratio, we describe this in more detail below.

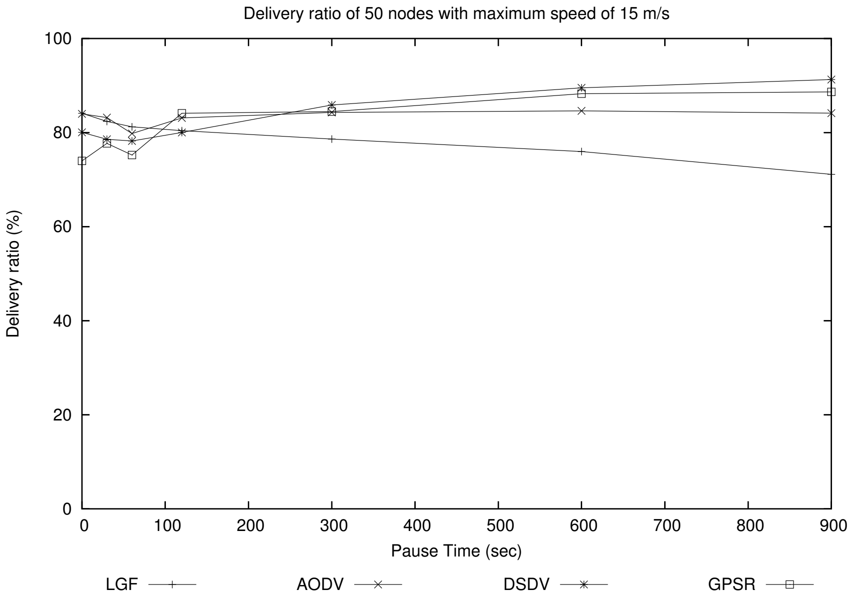

Figure 10: Comparison of the packet delivery ratio of the four routing protocols as a function of pause time with the maximum velocity set at 15 m/s

Both DSDV and AODV, upon notification of link retransmission failure, both protocols keep the packets in the buffer queue until route becomes available again. This techniques has not been published but it was found to be in the NS2 implementation. In the event of a link retransmission failure, GPSR applies the same technique used by DSR. It removes the routing entry of broken link before it en-queues the packet in the buffer for routing protocol to forward the packet to a different next hop[1]. In LGF, the protocol drops the packet, updates the route entry, and propagates the broken link to other neighboring nodes. Our results show that the link failure techniques used by GPSR, DSDV and AODV are opportunistic. The idea is to keep or redirect the packet when a link retransmission failure is encountered. Although this could increase the packet delivery ratio in some cases when connectivity are stable. However, in some scenarios such as where there is node mobility and the opportunity of direct or indirect re-delivery are not available, undelivered packets then linger for too long in the output buffer queue and can contribute to a higher average packet delay. Interestingly, our results show that other protocols gain an advantage in the scenarios which use pause times of 300, 600 and 900 seconds. Current

LGF design is unoptimized, we would expect to improve the performance of LGF in this respect.

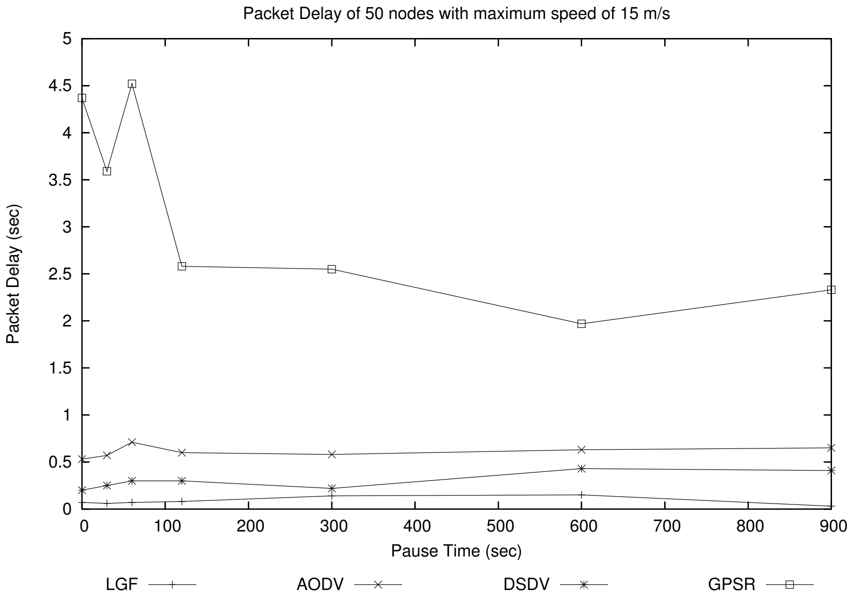

Figure 11: Comparison of average packet delay between the four routing protocols as a function of pause time with the maximum velocity set at 15 m/s

This optimization for increased packet delivery however does have side effects. From our observations, the average packet delay is increased as a results of this opportunistic delivery. In Figure 11, we show the effects on both AODV and DSDV are less significant as they only keep the undelivered packet for a short period of time. In contrast, GPSR retains the packet for much longer, this causes GPSR have an increased delivery ratio, but this has the side effect of a higher average packet delay. Our results show LGF consistently has a lower latency than other routing protocols. LGF achieves this by not holding the packets in the event of link retransmission failure.

Figure 12 highlights the communications overhead of the different routing protocols. In LGF, the node that is most likely to experience link failure within the next hop neighborhood advertises more frequently than nodes which are less likely to encounter link failure. Although, the advertisement is restricted to the local scope, LGF in general is sending out more frequent but restricted updates to its neighbors within its local scope. This explains why the overall communication overhead of LGF in this simulation is higher than DSDV. When compared with other protocols, LGF has a lower communication overheads than reactive AODV but higher overheads than DSDV or GPSR. Despite its merit of having a low routing overhead, GPSR can encounter the effect of stale state when connectivity to its adjacent nodes changes more rapidly its than neighbors' periodic advertisements.

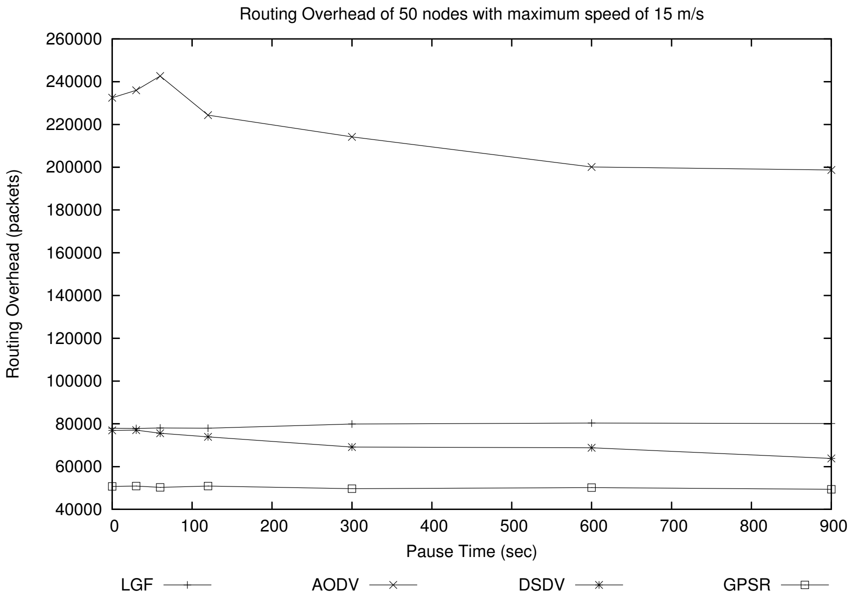

Figure 12: Comparison of routing overhead between the four protocols as a function of pause time with the maximum velocity set at 15 m/s

6.2 Performance with varying velocity

In this simulation, we tested performance of a system with 100 nodes over a wider area. Compared to previous simulations, the maximum distance between two nodes is larger, and therefore nodes are expected, on average, to take more hops between the source and destination. Additionally, the density of nodes in this simulation is 133 nodes per km^2 as compared to the previous density of 111 nodes per km^2. With more network overhead introduced as a result of the denser and larger system, it is further anticipated that contention and interference issues experienced in IEEE 802.11 networks could be more critical than previously measured. As a result, the channel capacity of the network is reduced[11] and consequently the average packet delay in general increases and the ratio of successful delivery decreases compared to previous simulations.

In comparison to other protocols, the results in figure 13 however does indicate that LGF is relatively steady and robust with respect to the measured delivery rate over a variety of velocities. We can conclude from these results that LGF is more reliable and adaptive to unsettled, dynamic topologies than other protocols.

Our results in figure 14 shows that LGF performs consistently well with respect to routing overhead over a variety of speeds. These results are similar to the previous simulation results, the high communication overheads associated with reactive AODV is a result of a higher number of route discoveries and local repairs AODV is performing. Comparing with earlier results where we used less nodes and smaller physical area, the overheads we observed are more onerous than in the previous simulation. Our observations show that the overheads associated with LGF are lower than the other protocol as the number of nodes is doubled from 50 to 100. Because DSDV needs to maintain global state for all the nodes in the network, its overheads increase in proportion to the number of nodes in the

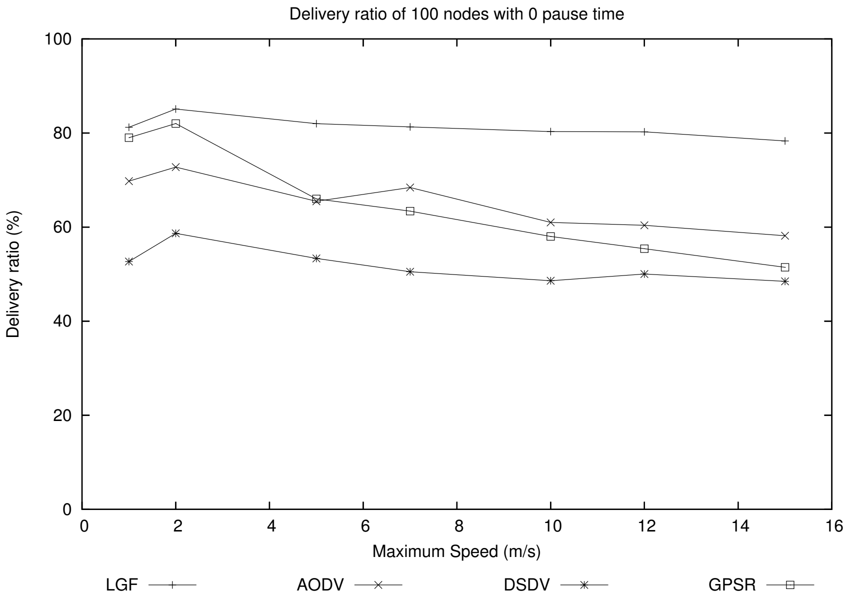

Figure 13: Comparison of packet delivery ratio between the four protocols as a function of maximum velocity where the pause time is zero

network. In the case of GPSR, every node advertises every 3 seconds which increases its overheads compared to DSDV where nodes advertise every 15 seconds. Although GPSR advertisements are only sent to its one hop neighbors, the results indicate that the higher frequency of GPSR updates can result in higher overheads as the network size increases than would be observed with DSDV in a similar scenario. In contrast the restricted route update in LGF adapts well to the increased size of the network with the results confirming LGF's communication overheads scale better than other protocols.

As shown in figure 15, DSDV does not converge fast enough to cope with the changes in connectivity when it uses a periodic update timer of 15 seconds when the network size has been increased. As a result of this, more undelivered packets are held in the queues in the network before they eventually expire and are dropped. Our results show the on demand path setup of AODV has a lower average packet delay than DSDV when simulating 100 nodes, this accounts for the performance advantage shown for the AODV local repair scheme in a dense network. If we considering the overall performance of all the protocol on packet delivery ratio, routing overheads and average packet delay, LGF provides a better overall balance performance than other protocols.

6.3 Path length

Figure 16 compares the path length for successful delivered packets for each protocol against the ideal shortest path retrieved from the NS2 simulator. The ideal shortest path is the shortest possible path only constrained by the physical radio range. The evaluation was carried out with a random way point mobility model using a 0 seconds pause time with a maximum velocity of 15 m/s and 50 nodes placed randomly in area of 1500x300 m^2.

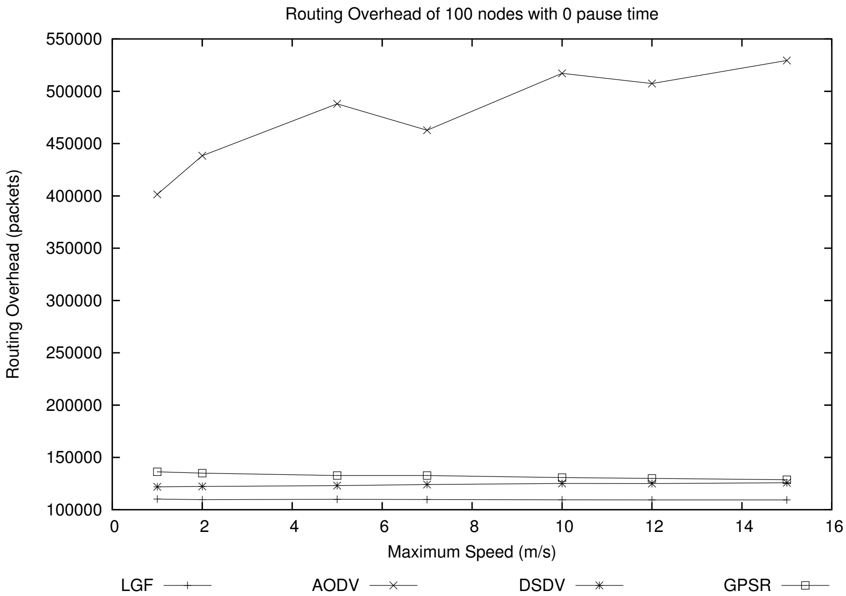

Figure 14: Comparison of the routing overheads between the four routing protocols as a function of maximum velocity where the pause time is zero

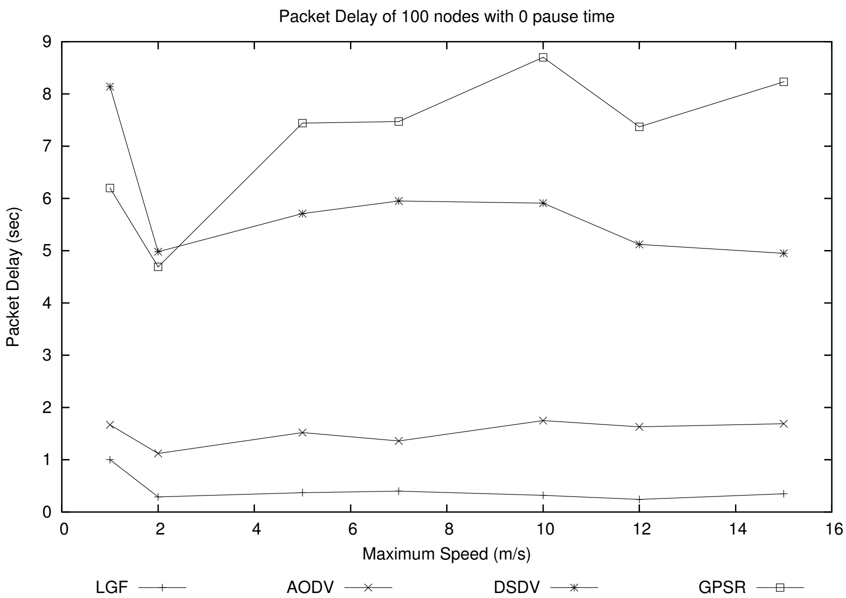

Figure 15: Comparison of average packet delay between the four routing protocols as a function of maximum velocity where the pause time is zero

The results indicate that LGF on average achieves 83.52 % of optimal path length while GPSR obtains 78.98 % of optimal path length. Although theoretically DSDV is supposed to maintain an optimal path, the slow update interval does not prevent misleading stale state from being used by the packet delivery mechanism and result in sub-optimal routing. DSDV only routes 77.37 % of its packets via the optimal path. Only 55.43 % percent of AODV's packets are routed by the optimal path. A contributing factor to this is AODV's local repair algorithm which is fixing broken paths without considering what the alternative optimal path between the source and destination is.

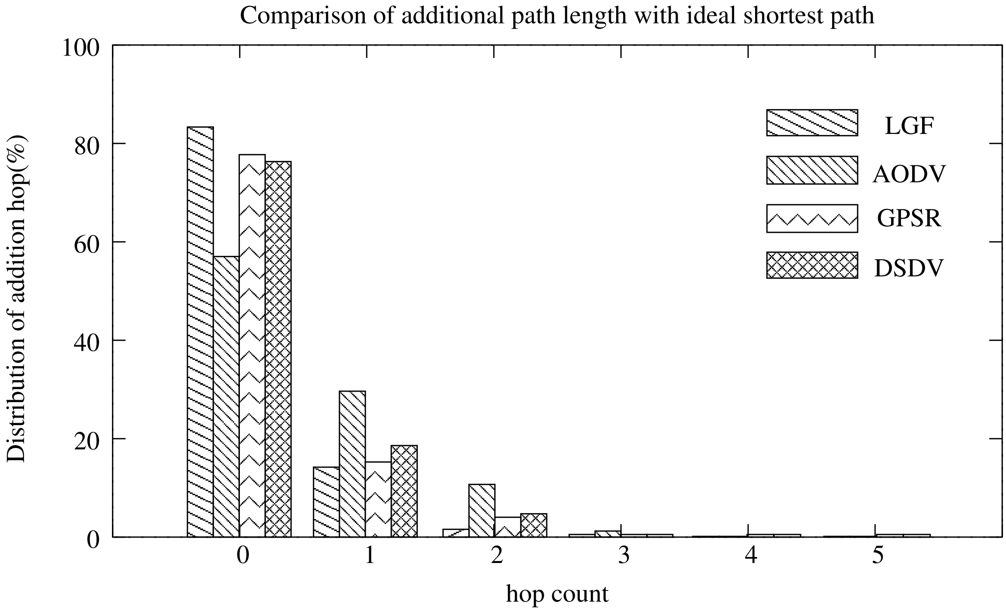

Figure 16: Comparison of the average path length for each of four protocols with ideal shortest path

7 Conclusion

In summary, we present a hybrid routing protocol, Landmark Guided Forwarding, using restrictive hybrid advertisement at a rate regulated by its connectivity sensitive algorithm, applying optimal routing when the destination is within its topological range, systematically resolves a transient next hop through local optimal resolution when an optimal route is unavailable.

We run simulations with 50 nodes and 100 nodes, the results indicate the overheads of LGF scale better than other protocols when the number of nodes is double from 50 to 100. In our performance evaluation with varying pause time, it is apparent that route optimizations by, AODV, DSDV and GPSR do improve the packet delivery ratio when rate of change of topology is low, using mobility model greater than 120 pause time at 15 m/s of maximum velocity, however the simulation results conclude these optimization could give side effect of higher reading in average packet delay. The effect is more pronounced when we simulated it at 100 nodes with slightly wider network diameter. In contrast, LGF is able to maintain a steady, swift and reliable delivery even with higher chance of unstable network connectivity. When comparing the path length with other protocols, LGF has the highest score of optimal routing than other protocols.

In conclusion, local optimal routing unlocks the constraint of maintaining a globally optimal path, as generally required by existing MANET protocols. LGF is therefore a relatively scalable and robust protocol with low overheads as compared to other Ad Hoc routing protocols.

8 Future work

8.1 Interface with internet coordinate scheme

In future work we wish to investigate using a coordinate system with the Landmark Guided Forwarding protocol to exploit their common goals of reducing routing overheads. The Internet coordinate systems such as Lighthouses [14] and Virtual Landmarks [17] could be used to supplement the process of selecting the topologically closest node to the destination. LGF could additionally exploit the topological data from the coordinate systems to avoid routing errors when removing edges or nodes that violate the triangle inequality.

8.2 Path optimization

In the current protocol the landmark node is chosen by its proximity to the destination, this normally results in sub-optimal routing that could be addressed by depositing additional soft-state information in the nodes. This information could include the paths learnt during previous packet deliveries and could be used to enable the path exploration algorithm used for other packets to exploit this knowledge while searching different destinations.

8.3 Resilient to position errors

Although we consider the handling of position inconsistencies while we designed the Landmark Guided Forwarding protocol, this feature is yet to be tested and fine tuned. This requires substantial work and thorough evaluation that has not been investigated in this paper. This investigation will form part of our future work.

References

[1] B.Karp and H.T.Kung. GPSR: Greedy perimeter stateless routing for wireless networks. *In Proceedings of 6th Annual Internation Conference on Mobile computing and Networking(MobiCom 2000), Boston, MA, USA*, pages 243–254, Feb 2000.

[2] Prosenjit Bose, Pat Morin, Ivan Stojmenovic, and Jorge Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, 7(6):609–616, 2001.

[3] C.Perkins and P.Bhagwat. Highly dynamic destination-sequenced distance-vector routing(DSDV) for mobile computing. *Computer Communication Review*, 24(4):234–244, Oct 1994.

[4] D.Johnson, D.A.Maltz, and J.Broch. *DSR: The Dynamic Dource Routing Protocol For Multihop Wireless Ad Hoc Networks, Ad Hoc Networking.* Addison-Wesley Longman Publishing Co. Inc., Boston, MA, 2001.

[5] NS group at ISI. Ns 2 home page. World Wide Web, `http://www.isi.edu/nsnam/ns/`.

[6] Zygmunt J. Haas and Marc R. Pearlman. *ZRP: a hybrid framework for routing in Ad Hoc networks. Book chapter in Ad Hoc Networks, Editor C. Perkins.* Addison-Wesley Longman Publishing Co., Inc., 2001.

[7] Yongjin Kim, Jae-Joon Lee, and Ahmed Helmy. Modeling and analyzing the impact of location inconsistencies on geographic routing in wireless networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 8(1):48–60, 2004.

[8] Antony LaMarca, Yatin Chawathe, Sunny Conolvo, Jeffery Hightower, Ian Smith, James Scott, Tim Sohn, James Howard, Jeff Hughes, Fred Potter, Jason Tabert, Pauline Powledge, Gaetano Borriello, and Bill Schilt. Place lab: Device positioning using radio bacons in the wild. Technical report, Intel Research, 2004.

[9] J.Yves L.Blazevic and S.Ciordano. A location-based routing method for mobile ad hoc networks. *IEEE Transaction on Mobile Computing*, 3(4):243–254, Feb 2000.

[10] J. Li, J. Jannotti, D. De Couto, D. Karger, and R. Morris. A scalable location service for geographic ad hoc routing. *In Proceedings of the 6th ACM International Conference on Mobile Computing and Networking (MobiCom '00)*, pages 120–130, August 2000.

[11] Jinyang Li, Charles Blake, Douglas S. J. De Couto, Hu Imm Lee, and Robert Morris. Capacity of ad hoc wireless networks. *Mobile Computing and Networking*, pages 61–69, 2001.

[12] Elizabeth M.Royer and C. K. Toh. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications Magazine*, pages 46–55, April 1999.

[13] C. Perkins and Elizabeth M. Royer. Ad hoc on demand distance vector routing. *In Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA.*, pages 90–100, February 1997.

[14] M. Pias, J. Crowcroft, S. Wilbur, S. Bhatti, and T. Harris. Lighthouses for scalable distributed location. *In Second International Workshop on Peer-to-Peer Systems (IPTPS '03)*, Feb 2003.

[15] V.R Syrotituk S.Basagni, I Chlamtac and Woolward. A distance routing effect algorithm for mobility(dream). *In Proceedings of ACM/IEEE Internation Conference on Mobile Computing and Networking(Mobicom 98)*, pages 76–84, 1998.

[16] Charles E.Perkins S.Dass and Elizabeth M.Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. *IEEE Personal Communication Magazine, special issue on mobile Ad Hoc networks*, 8(1):16–29, Feb 2001.

[17] L. Tang and M. Crovella. Virtual landmarks for the internet. *In Proceedings of Internet Measurement Conference 143–152, Miami Beach, FL.*, October 2003.