GeoTORA: A Protocol for Geocasting in Mobile Ad Hoc Networks *

Young-Bae Ko and Nitin H. Vaidya

Department of Computer Science
Texas A&M University
College Station, TX 77843-3112, USA
E-mail: {youngbae,vaidya}@cs.tamu.edu
http://www.cs.tamu.edu/faculty/vaidya/mobile.html

Technical Report 00-010

March 6, 2000

Abstract

In this report, the problem of providing a geocast service in mobile ad hoc networks is considered and a novel geocasting algorithm combining unicast and flooding presented. Geocast is useful for sending messages to everyone in a specified geographical region. The proposed protocol is named *GeoTORA* because it is derived from the TORA (unicast) routing protocol. Flooding is also incorporated in GeoTORA, but it is limited to nodes within a small region. This integration of TORA and flooding can significantly reduce the overhead of geocast delivery, while maintaining high *accuracy*.

1 Introduction

Geocasting has been proposed as a mechanism to deliver messages of interest to all hosts within a given geographical region [20]. In traditional multicasting, a host becomes a member of the multicast group by explicitly joining the multicast group. On the other hand, a host automatically becomes a member of a geocast group if its location belongs to the region specified for the geocast. Imielinski and Navas [20] proposed approaches for geocasting in the internet. Ko and Vaidya [17] have proposed the use of geocasting in mobile ad hoc networks, and presented a protocol based on flooding. This paper improves on the algorithm presented by Ko and Vaidya [17]. Since we compare our results with those for the algorithm proposed in [17], this paper uses terminology and performance metrics similar to [17].

As noted above, geocasting allows delivery of packets to nodes within a specified geographical region – this region is referred to as the *geocast region* [17]. The set of nodes in the geocast region is said to form the *geocast group*. For a node to be able to determine whether it belongs to a geocast group or not, the node should be able to know its own physical location. A node can determine its location, for instance, using the Global Positioning System (GPS).

This report is organized as follows. Section 2 summarizes related work. Section 3 presents a general description of GeoTORA, without presenting any implementation details. A more detailed description of TORA and GeoTORA is presented in Sections 4 and 5. Performance evaluation results are discussed in Section 6. Section 7 presents our conclusions.

*Research reported here is supported in part by National Science Foundation grants ANI-9976626 and ANI-9973152.

2 Related Work

This report presents a new algorithm for geocasting in mobile ad hoc networks. A mobile ad hoc network consists of mobile hosts that communicate with each other over wireless links [9]. In a mobile ad hoc network, typically, all mobile hosts behave as routers. A route between a pair of nodes in a mobile ad hoc network may go through several other mobile nodes. These routes can change when hosts[1] change location. Therefore, there has been significant research on the development of (unicast) routing protocols for mobile ad hoc networks [1, 2, 4, 8, 12, 14, 16, 18, 23, 24, 26, 27]

In addition to the above work on unicast routing in ad hoc networks, there has been significant work on multicasting as well, and several approaches have been proposed [3, 5, 6, 7, 11, 13, 15, 19, 21, 25, 28]. The schemes for multicasting can be broadly divided into two types: *flooding*-based schemes and *tree*-based schemes. Both approaches have their advantages and disadvantages. Flooding-based schemes do not need to maintain as much network state as the tree-based protocols. On the other hand, flooding-based schemes can potentially deliver the multicast packets to a large number of nodes who do not wish to receive them (i.e., nodes which do not belong to the multicast group). Tree-based schemes tend to avoid this drawback of flooding-based schemes, at the cost of increased overhead in tree maintenance.

As noted earlier, the concept of geocasting was introduced by Imielinski and Navas [20]. They also presented an architecture to implement geocasting in the internet. Recently, Ko and Vaidya [17] presented the so-called *Location-Based Multicast* (LBM) algorithm that uses flooding to deliver a geocast packet. However, to reduce the propagation of the flood, LBM limits the flood to a *forwarding zone* – the forwarding zone covers a subset of the network, and is determined based on the location of the sender and coordinates of the geocast region. Although the algorithm in [17] is able to limit the flood of geocast packets to a relatively small region, still many nodes outside the geocast region tend to receive the geocast packet.

3 A General Description of GeoTORA

Since GeoTORA is based on TORA, we begin with an abstract description of TORA.

3.1 TORA Protocol

TORA (Temporally Ordered Routing Algorithm) [23] is one of a family of *link reversal* algorithms [10] for routing in ad hoc networks. For *each* possible destination in the ad hoc network, TORA maintains a *destination-oriented* directed acyclic graph. In this graph structure, starting from any node, if links are followed in the logical direction of the links, the path leads to the intended destination. TORA uses the notion of *heights* to determine the direction of each link – we will discuss this in greater detail later. Despite dynamic link failures, TORA attempts to maintain the destination-oriented DAG such that each node can reach the destination, as illustrated below.

Figure 1 illustrates how link reversal is performed in TORA. An arrow connecting a pair of nodes in this figure implies that the two nodes can communicate with each other. That is, the physical link between the two nodes is bidirectional. However, the TORA algorithm imposes a *logical* direction on the links, as illustrated in Figure 1(a) – this figure shows the destination-oriented DAG with node G being the destination. Observe that, starting from any node in the graph, the destination G can be reached by simply following the directed links.

Now assume that the link between nodes D and F breaks (perhaps because node F moves away from node D). Then, in the destination-oriented DAG, node D does not have any outgoing logical link. In response, TORA *reverses* logical direction of the (D,B) and (D,C) links, as shown in Figure 1(c). Now, node C does not have any outgoing logical link. In response, logical direction of link (B,C) is reversed,[2] resulting in the graph in Figure 1(c). Now since node B does not have any outgoing logical link, the logical direction of link (A,B) is reversed, resulting in the destination-oriented DAG in Figure 1(d). In this state, each node (other than the

[1] We will use the terms *node* and *host* interchangeably.

[2] TORA employs a partial reversal algorithm. Thus, only some or all incoming links at a node may be reversed.

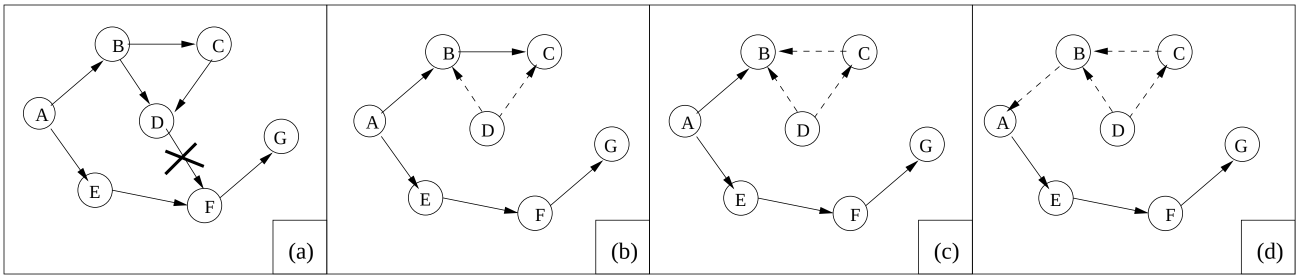


Figure 1: Route Maintenance in TORA: Route maintenance is required due to failure of the link between nodes D and F. In the figure, a link that has been reversed since the initial state is shown as a dashed line.

destination G) has an outgoing logical link, and is able to reach the destination node G by following the directed links.

3.2 Anycasting Using Modified TORA

To implement GeoTORA, we first modify TORA to be able to perform *anycast*. To perform an *anycast*, an anycast group is defined – anycast group consists of a subset of the nodes in the network. When a node sends a message to the anycast group, the message is delivered to any one member of the anycast group.

While TORA maintains a DAG for each destination, the anycasting algorithm would maintain a *single* DAG for a given *anycast group*. Observe that, in steady state, when using TORA, only the intended destination node is a *sink* node in its destination-oriented DAG. To perform anycasting, we modify TORA to maintain a DAG structure such that all nodes belonging to the anycast group are sinks. In this case, a link that is between two nodes belonging to anycast group is not given a logical direction.

Figure 2 illustrates the anycast scheme. In this case, let us assume that nodes A, B, C and D belong to the anycast group. The present DAG structure is shown in Figure 2(a). Observe that links with both endpoints in the set {A,B,C,D} do not have any logical direction. From any node that is outside the anycast group, following the directed links leads to one member of the anycast group. Now, suppose that node G moves, breaking link (A,G) – the resulting DAG structure is shown in Figure 2(b). Observe that now node G does not have any outgoing link. In response, the logical direction of link (G,J) is reversed, resulting in the DAG shown in Figure 2(c). Now all nodes that are outside the anycast group have an outgoing link (and a path to at least one node in the anycast group).

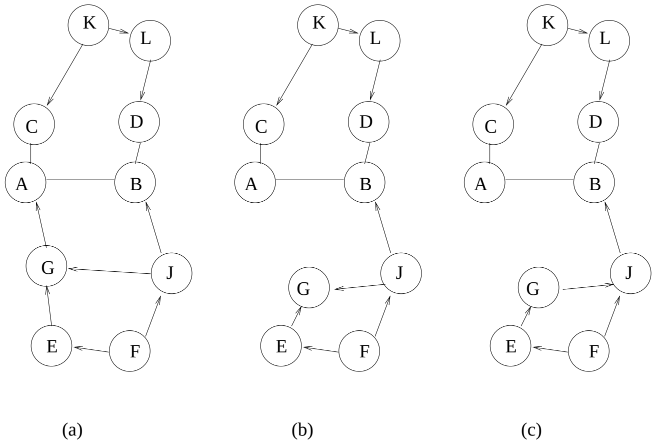


Figure 2: Anycasting using modified TORA

3.3 GeoTORA Protocol

The GeoTORA protocol is obtained using a small variation on the above anycasting protocol. Consider the system shown in Figure 3(a). In this case, let us assume that the circle represents the geocast region. Thus, the geocast group at the present time is the set of nodes {A,B,C,D}. GeoTORA maintains a single DAG for each geocast group – the DAG is updated when membership of the geocast group changes.

To perform geocasting using GeoTORA, first, a sender node essentially performs an anycast to the *geocast* group members – similar to the above anycast protocol, in GeoTORA, logical directed links are set up such that a node wishing to perform a geocast can reach any one node in the geocast group by simply forwarding the packet on any outgoing link. When any node in the geocast group receives the packet, it *floods* the packet such that the flooding is *limited to the geocast region* – to achieve this, only nodes that are within the geocast region (i.e., the geocast group members) forward the flooded packet; other nodes simply drop the flooded packet. To ensure that a given node does not forward a flooded packet more than once, a sequence number is attached to each packet, similar to the flooding schemes used in other protocols [16, 18].

For instance, if node E in Figure 3(a) wants to perform a geocast, it forwards the geocast packet to node G, along the outgoing link (E,G). Node G, in turn, forwards the packet to node A. Since node A is in the geocast region, it initiates a *flooding* of the packet *limited to the geocast region*. Nodes B and C, on receiving the packet from node A, forward the packet to their neighbors. When node A receives the packet from node B or C, it does not forward the packet, since node A has already once forwarded the packet to its neighbors. In this manner, the packet will reach nodes A, B, C and D that belong to the geocast region.

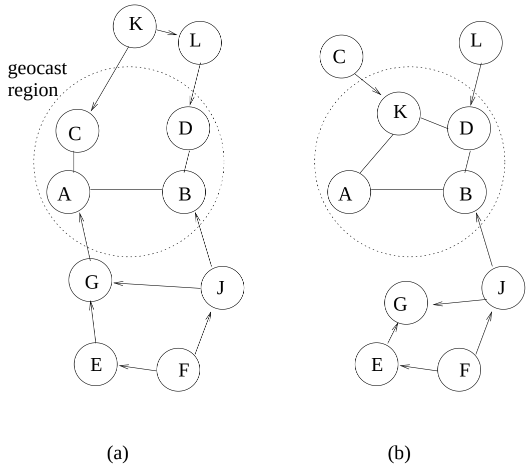


Figure 3: Geocasting using GeoTORA

Since mobile hosts may move into and out of the geocast region, the set of nodes in the geocast group can change dynamically. Thus, we need to incorporate mechanisms to allow a node that is not a *sink* (i.e., a group member) to become a sink, and vice versa. GeoTORA incorporates such mechanisms, as described in more detail in the next section. Here we explain the behavior of the protocol by continuing with our example.

Again, consider Figure 3(a). Now suppose that node C moves out of the geocast region, and node K moves into the geocast region simultaneously. The resulting DAG (after GeoTORA takes appropriate steps) is shown in Figure 3(b). Observe that node K has now become a sink, and node C is no more a sink. Since node K has moved closer to node A, a link exists between nodes A and K. However, this link is not given a logical direction, since nodes A and K are now both in the geocast region.

There are two other possibilities that need to be handled in GeoTORA: (a) GeoTORA needs to handle the case when all the geocast members may be partitioned from some of the other nodes in the network. (b) GeoTORA also needs to handle the case when the geocast region is empty. In this case, eventually, some node may again enter the geocast region. Thus, the transitions between a non-empty geocast group and an empty geocast group must be considered.

This section presented an abstract description of the GeoTORA protocol. A discussion of additional details is presented in Section 5.

4 Temporally-Ordered Routing Algorithm (TORA) [23]

As GeoTORA is based on TORA, we first present some more details of TORA. Readers familiar with the details of TORA may omit this section without loss of continuity. TORA, proposed by Park and Corson, provides loop-free, (potentially) multiple routes from any source to a desired destination. In order to forward data packets to a given destination, a node simply needs to maintain direction for its links. Logical direction of a link between a pair of nodes is determined by assigning a *height* to each node. The logical links are considered to be directed from nodes with higher height towards nodes with lower height – lexicographic ordering on height is used since height is defined as a five-tuple, $(\tau, oid, r, \delta, i)$, as explained below. The height consists of two components: a *reference level* represented by the first three components of the five-tuple, and a "delta" with respect to the reference level, represented by the last two components of the five-tuple. Each component of the five-tuple is explained below:

- τ: A new reference level is defined each time a node loses its last outgoing link. τ is a tag that represents the time of the link failure.
- *oid* (originator id): Unique identifier of the node that defined the new reference level. The *oid* ensures that the reference levels can be totally ordered lexicographically even if multiple nodes define reference levels simultaneously.
- r: Reflection indicator bit. This bit is used to detect the fact that a partition has occurred in the network. This bit is initially set to 0. As seen earlier, when a node does not have any outgoing links as a result of a link failure, it reverses some (or all) of its incoming links. The reaction to a link failure propagates through other nodes that have lost all their routes to the destination, as a result of the link failure. When the reaction propagates to a node that originally had only outgoing links, but now has no outgoing links (since all the outgoing links were reversed by its neighbors), the node "reflects" the link reversals, by setting its height higher than any of its neighbors. The r bit is used for this purpose.
- δ: Propagation ordering parameter. The use of this parameter will be explained by means of an example below.
- i: Unique node identifier (ID)

TORA performs three basic functions: route creation, route maintenance, and route erasure. Three control packets – query (QRY), update (UPD), and clear (CLR) – are used to accomplish these functions. Creating routes from various sources to the destination corresponds to establishing a sequence of directed links from each source to the destination. This is accomplished by maintaining a directed acyclic graph (DAG) rooted at the destination. A query/reply process with QRY and UPD packets is used for building the destination-oriented DAG. Figure 4 illustrates the process of route creation, with time increasing from Figure 4(a) to Figure 4(f).

Initially, as shown in Figure 4(a), height H_i of each node i (other than the destination) is set to NULL – specifically, $H_i = (-, -, -, -, i)$. Note that although the last component in $(-, -, -, -, i)$ is not null, the height is considered to be equal to NULL. The destination node G sets its height to be ZERO $= (0, 0, 0, 0, G)$. Now, when any node with no outgoing links (for instance, node A in Figure 4(a)) requires a route to the destination (node G in Figure 4), it broadcasts a QRY packet to all of its neighbors and sets a *route-required* flag. When a node, say X, receives a QRY packet, it reacts in accordance with the following rules:

a) If node X has no downstream links and its *route-required* flag is un-set, then it just forward the QRY to neighbors, while setting the route-required flag. Note that a link between two nodes whose height is NULL is considered *undirected*. On the other hand, a NULL height is considered to be higher than any non-NULL height.

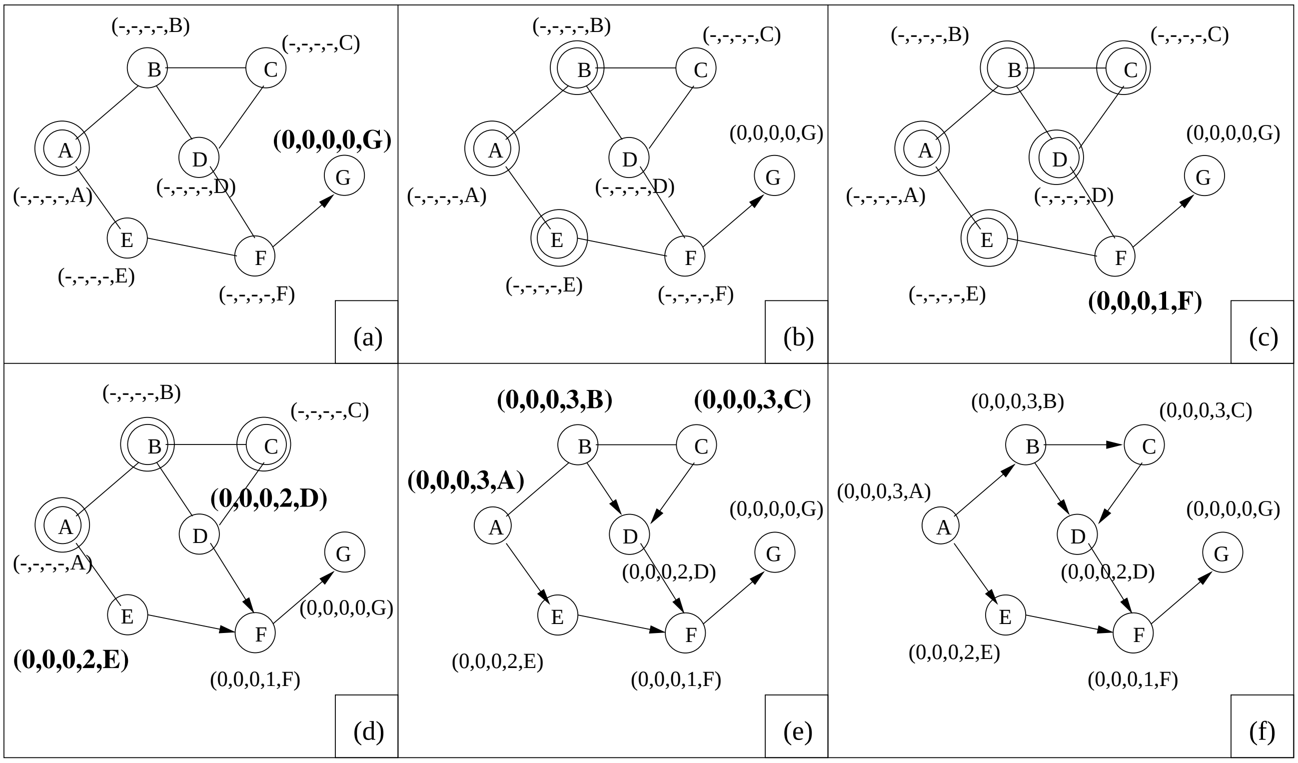


Figure 4: Route Creation Phase in TORA: In the figure, a circle around a node indicates that the route-required flag is set. Arrows on each wireless link points from the higher height node to the lower height node. The height is depicted as a 5-tuple, as explained in the context.

Observe that nodes B and E in Figure 4(b) apply this rule on receiving a QRY from node A, and forward the QRY packet to their neighbors. In Figure 4, a double circle around a node indicates that the *route-required* flag is set at that node.

b) If node X has no downstream links, but its route-required flag is set, then node X simply discards the received QRY packet. For example, when node A receives a QRY from nodes B or E, it will drop the QRY without any further reactions (since node A's route-required flag was set when it forwarded the query to its neighbors).

c) If node X has at least one downstream link, and its height is NULL, it modifies value of δ in its height, based on the relative height metric of neighboring nodes. Thus, node X changes its current NULL height (-,-,-,-,X) to (τ,oid,r,δ+1,X), where (τ,oid,r,δ,i) is the minimum height of its non-NULL neighbors (this height corresponds to some neighbor node, i). Also, node X sends an UPD packet to its neighbors containing the new height.

   In our example, in Figure 4(c), node F updates its height to $(0, 0, 0, 1, F)$, since the only non-NULL height among its neighbors is height $(0, 0, 0, 0, G)$ of the destination node G. Node F, then, transmits an UPD to its neighbors.

d) If node X has at least one downstream link but its height is non-NULL, it first compares the time the last UPD packet was broadcasted with the time when the link over which the QRY packet was received became active. If an UPD packet has been broadcasted since the link became active, it simply discards the QRY; otherwise, node X broadcasts an UPD packet as a response, to inform its height to neighbors.

   For instance, in Figure 4(d), node F may receive a QRY packet from node D after node F has sent an UPD packet, as seen earlier (with reference to Figure 4(c)). This results in node F discarding the QRY packet from node D.

When a node, say Y, gets an UPD packet from its neighbor, node Y checks its route-required flag to see if it is set or not. If the flag is set (meaning that the height of node Y is NULL), then it updates its height as (τ,oid,r,δ+1,Y), based on the minimum height value (τ,oid,r,δ,i) of its non-NULL neighbors. Node Y then broadcasts an UPD containing its new height. On the other hand, if the route-required flag of node Y is unset, Y need only react if it has lost its last downstream link. In Figure 4(e), nodes B and C update their heights in this manner. In turn, node A updates its height as $(0, 0, 0, 3, A)$ since its route-required flag was set and its non-NULL neighbors' minimum height is $(0, 0, 0, 2, E)$ when it receives an UPD from node E. When route creation process initiated by node A in Figure 4 completes, the heights of the nodes along any route to the destination are strictly decreasing, as shown in Figure 4(f).

Destination-oriented DAG established by the route creation process can break due to a link failure. In this case, a procedure for maintaining routes is necessary in order to rebuild the DAG rooted at the destination. TORA does not react to link failures as long as there are still outgoing links available at each node (other than the destination). If some node, say node Z, loses all its outgoing links, then it reverses the direction of some or all of its incoming links. Link reversal is performed by choosing a new *reference level* for the height such that the height of node Z becomes higher than any other node in the network. The node that chooses a new reference level then broadcasts an UPD packet containing the new reference level to its neighbors. If such a link reversal by node Z causes another node (say, node W) to lose its last downstream link, node W adjusts its height to be "lower" than the height of the sender of the UPD, i.e., node Z, and broadcasts an UPD. This process of link reversal[3] continues until either all nodes have at least one downstream link (See Figure 1) or a network partition is detected.

One of advantages in TORA protocol is the fact that a network partition can be detected during the route maintenance phase. This capability leads to a procedure for erasing routes. In the route erasure process, a CLR control packet is flooded throughout the network to erase all invalid routes so that all links of nodes partitioned from the destination become undirected.

Due to lack of space, it is not possible to illustrate all details of TORA with sufficient clarity. The readers are referred to [23] for further explanations.

[3]Note that TORA is based on a partial link reversal algorithm. As an example, in Figure 1(c), node C only reverses the link from node B to node C, but not the link from node D to node C.

5 Proposed GeoTORA Protocol

We now further elaborate on GeoTORA. Since GeoTORA is quite similar to TORA, we primarily highlight the differences between TORA and GeoTORA in this section. First, the route creation and maintenance in GeoTORA is discussed, followed by the procedure for delivery of geocast messages using GeoTORA. Recall that, as discussed in Section 3, for each geocast group, GeoTORA maintains a single directed acyclic graph (DAG). This is similar to the DAG maintained by TORA, with the difference being that *all nodes* that belong to the geocast region have a ZERO height – link between a pair of nodes is not assigned a direction if both nodes have ZERO height. This is unlike TORA, where only a single node (the destination) has ZERO height.

5.1 Route Creation and Maintenance in GeoTORA

In order to deliver packets to the geocast group, a source should have a route to the given geocast region. To establish routes initially, GeoTORA uses a route *creation* process that is essentially identical to that for TORA, but with the difference noted above (i.e., all geocast members have ZERO height). Figure 5 provides an illustration for the process of geocast route creation in GeoTORA.

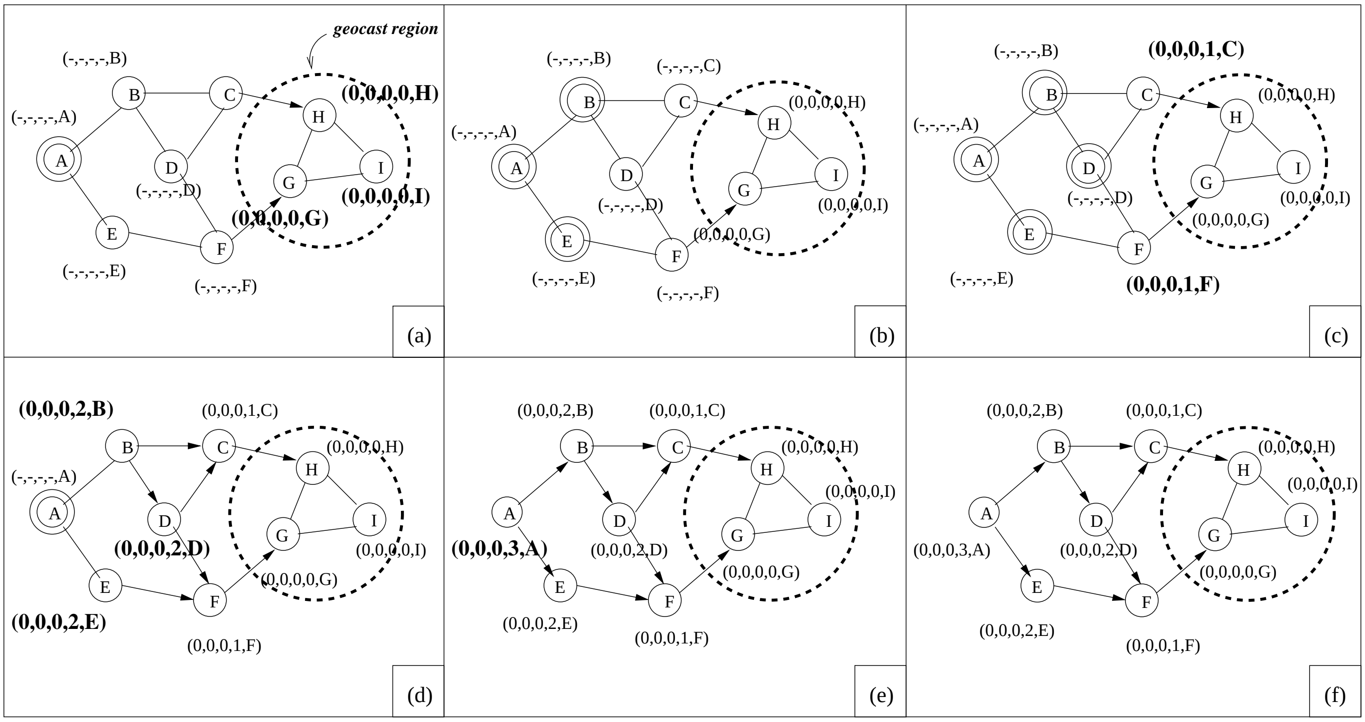


Figure 5: Geocast Route Creation in GeoTORA

In Figure 5, the dotted circle represents the geocast region – nodes G, H and I are within the geocast region (in this example, the set of nodes in the geocast region does not change). Figure 5(a) represents the initial state of the system. Since nodes G, H and I are within the geocast region, they set their height to ZERO. Any other node, say i, sets its height to NULL – specifically, node i sets its height to be $(-,-,-,-,i)$. Note that links between two nodes with ZERO height are not assigned any direction; similarly links between two nodes with NULL height are also not assigned any direction. Nodes C and F (whose height is NULL) have links, respectively, to nodes H and G (whose height is ZERO). Therefore, the links (C,H) and (F,G) are assigned a direction – recall that NULL height is considered to be greater than any non-NULL height. Rules for route creation process in GeoTORA are identical to those described for TORA.

Assume that node A wishes to perform a geocast to the geocast group. Since node A does not have any outgoing link, it transmits a QRY packet to its neighbors, and sets its *route-required* flag. Note that in Figure 5, a double circle around a node indicates that the route-required flag at that node is set. The QRY packet transmitted by node A reaches nodes B and E, and they, in turn, forward the packets to their neighbors, and also set the local route-required flag (refer Figure 5(b)). Nodes C and D receive the QRY message from node B, and node F receives from node E. In Figure 5(c), observe that nodes C and F have outgoing links to geocast group members, but node D does not. Therefore, only node D forwards the packets to its neighbors, and sets its route-required flag. On the other hand, on receiving a QRY, nodes C and F change their height from NULL to $(0,0,0,1,C)$ and $(0,0,0,1,F)$, respectively, and send UPD message to their neighbors informing the new height. Response of a node on receiving an UPD message is identical to that in TORA. Figures 5(d) through (f) show evolution of the algorithm beyond the stage shown in Figure 5(c). At the end, as seen in Figure 5(f), a DAG is established wherein each geocast group member is a sink.

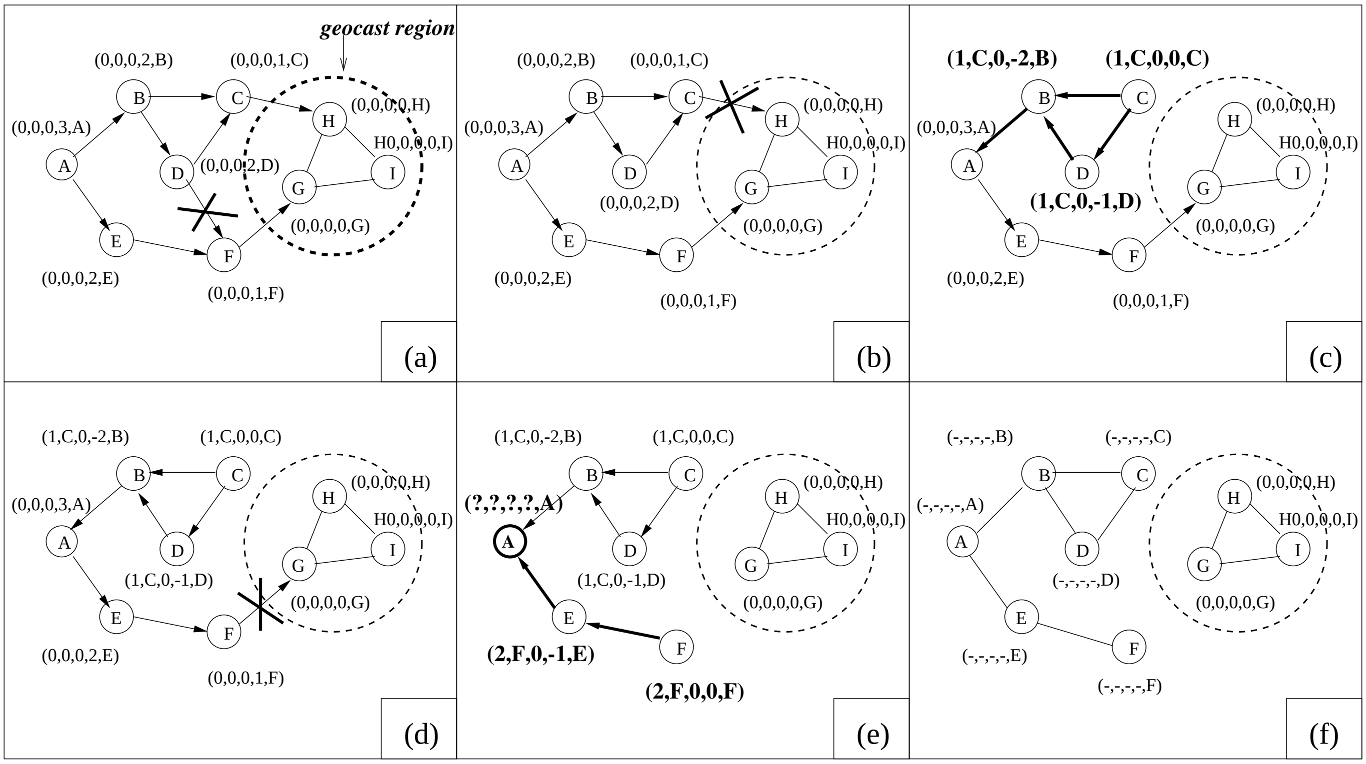


Figure 6: Geocast Route Maintenance in GeoTORA: Three different scenarios of link failure

Now we illustrate route maintenance in GeoTORA. In GeoTORA, the DAG may need to be modified when: (a) a link failure occurs, or (b) when a node enters or leaves the geocast region.

The GeoTORA route maintenance procedure in response to link failures is similar to TORA. Figure 6 illustrates how the DAG is modified in GeoTORA in response to link failures, considering several link failure scenarios. Figure 6(a) shows the case where no maintenance reaction is taken by node D, as a result of breakage of link (D,F), since node D still has an outgoing link (D,C). Next, as shown in Figure 6(b), the link from node C to node H breaks. Now, node C is left without any outgoing links – let us assume that the link failure occurred at time 1. Node C then updates its height using a new reference level representing the fact node C has lost all downstream links at time 1. The new height of node C is *(1,C,0,0,C)*, as shown in Figure 6(c). Node C also generates an UPD containing its new height and broadcasts the UPD to neighbor nodes – the procedure for handling the UPD messages in GeoTORA is identical to TORA. Since node C increases its height, now node D also has no outgoing links – in response, node D chooses height $(1,C,0,-1,D)$ and sends an UPD to its neighbors. The new height chosen by nodes C and D results in the loss of the only outgoing link from node B. Therefore, node B lowers its height to $(1,C,0,-2,B)$, to be lower than the current height of node D, and transmits an UPD containing its new height. The new height chosen by node B causes reversal of the link between node A and B to, now, point to node A. However, node A still has another outgoing link (A,E), so no further action is needed. The final state of the DAG, after the failure of the link between nodes C and H is shown in Figure 6(c).

Sometimes, a link failure causes a network partition, such that some nodes may not have any path remaining to any node in the geocast group. For instance, Figure 6(d) depicts the case where link between nodes F and G is broken. Now, assume that the time when failure occurred is 2. The reaction to this link failure is similar to the reaction following failure of link (C,H) in Figure 6(b). As a result of the failure of link (F,G), nodes F and E choose new height. The resulting state is shown in Figure 6(e). Observe that, before the link failure, node A only had outgoing links (no incoming links). Now, however, due to the new height chosen by node E, node A has no outgoing links remaining. Node A realizes that all its outgoing links are broken when it receives an UPD message from node E containing node E's new height. Subsequently, following the "reflection" procedure as defined in TORA, the fact that A is partitioned from the geocast group is detected. Therefore, route erasure phase is initiated. Details of the route erasure phase are not illustrated here for brevity – however, note that the procedure is identical to TORA. Figure 6(f) shows the network state after route erasure process has been completed. Until a new route to the geocast region is detected, a source that is partitioned from the geocast group is not able to send geocast data packets.

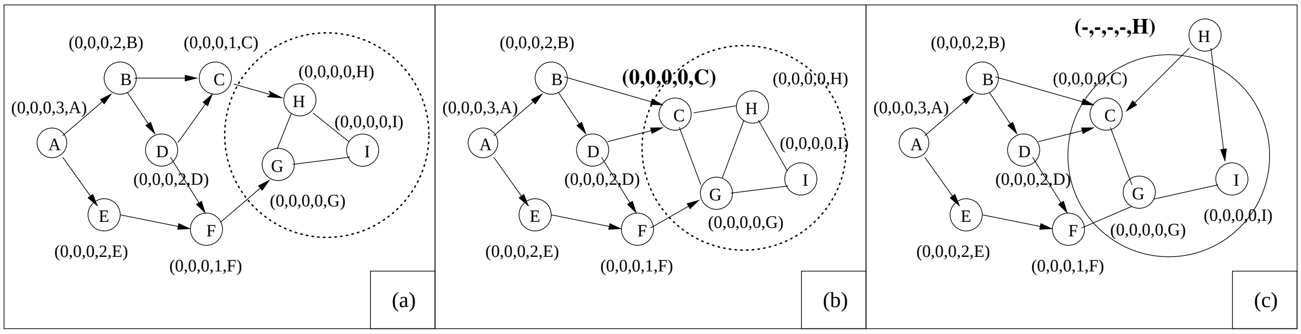


Figure 7: Route Maintenance in GeoTORA for handling dynamic change of geocast group

Figure 7 illustrates how GeoTORA handles geocast group membership changes. Consider Figure 7(a) as an example network. In Figure 7(a), when node C moves into the geocast region and becomes a sink, it simply updates its current height to be ZERO. And a UPD is transmitted by node C to inform its new height to its neighbors. The resulting state is shown in Figure 7(b). Now, let us assume that node H leaves the geocast group by moving out of the geocast region. In this case, the height of node H is set to NULL. Note that a NULL height is considered greater than any non-NULL height. Therefore, undirected links (H,C) and (H,I) in Figure 7(b) now have logical directions from node H to C and from H to I, respectively, as shown in Figure 7(c).

5.2 Delivery of Geocast Packets

Geocast delivery using GeoTORA consists of two phases: *anycasting* phase, and *local flooding* phase, as discussed below.

Anycasting Phase

When a node wishes to send a packet to the geocast group, it forwards the packet on any of its outgoing links[4]. Each node that receives the packet forwards the packet on an outgoing link. Provided the source node is not partitioned from the geocast group, the packet eventually reaches *one member* of the geocast group.

[4]If no outgoing link is available, then the appropriate steps in route creation and maintenance procedures are first invoked.

Local Flooding Phase

Once a packet is delivered to one node in the geocast group (by the anycasting phase above), that node initiates *local flooding* of the packet. The purpose of local flooding, described below, is to deliver the packet to the remaining geocast group members. The node, say X, that initiates the flood, tags the specification of the *geocast region* to the packet, and broadcasts it to its neighbors. Any node, say Y, that receives the flooded packet verifies whether it is within the region whose specification is tagged to the packet. If node Y is outside the region, then it simply discards the packet. On the other hand, if node Y is within the tagged region, then node Y broadcasts the packet to its neighbors. Caution is taken to ensure that a given node would not broadcast the same packet more than once.

Local flooding initiated by node Y may not necessarily deliver the packet to *all* nodes within the *geocast region*. Particularly, using the above procedure, the packet would not be delivered to a geocast group member, say Z, if there is no path from Y to Z that consists of nodes belonging to the geocast region only (since local flooding above is confined to the geocast region). The probability that the packet would be delivered to all the geocast group members can be increased by using a larger region for the local flooding. There exists a trade-off between the overhead of local flooding and the number of geocast group members who receive the packet.

6 Performance Evaluation

For the evaluation purpose, the proposed GeoTORA protocol is compared to *pure geocast flooding* and the *Location-Based Multicast (LBM)*[5][17] algorithms. Pure geocast flooding floods the whole network, and LBM scheme limits the flooding to the smallest rectangular region containing a source node and a geocast region. We performed a simulation study using an extended version of the network simulator *ns-2* [4]. *ns-2* is a discrete event-driven network simulator with extensive support for simulation of TCP, routing, and multicast protocols. The extensions implemented by CMU Monarch Project were used for our simulations. Their extensions enable simulation of multi-hop wireless ad hoc networks. Extensions include simulation modules for the IEEE 802.11 MAC layer protocol and a radio propagation model.

6.1 Simulation Model

In our simulation model, initial locations (X and Y coordinates) of the nodes are obtained using a uniform distribution. The nodes, chosen to be 30 nodes, move around in a rectangular region of size 700 unit x 700 unit square according to the following mobility model: each node chooses a direction, moving speed, and distance of move based on a predefined distribution and then computes its next position P and the time instant T of reaching that position. Each node moves with three different *maximum* speeds: 5, 10 and 20 units per second (i.e., average speeds of 2.5, 5 and 10 units per second respectively). We ran our simulations with movement patterns generated for 4 different pause times: 0, 10, 100, and 1000 seconds. A pause time of 0 seconds corresponds to continuous motion.

Two mobile hosts are considered disconnected if they are outside each other's transmission range, which is defined as 250 units for all nodes. The wireless link bandwidth is 2 Mbps. One of the nodes is chosen as the sender for the geocasts – it initiates a geocast. In our simulation, simulation time is *inversely* proportional to the speed. For instance, simulations for the maximum speed of 5 units per second run 1000 seconds of execution, whereas run for 500 seconds for maximum speed of 10 units per second.

For the simulations, any data packets that cannot be delivered due to a broken route are simply dropped. The size of data payload is 512 bytes. Unless otherwise stated, 1000 geocasts have been done in each simulation run. For GeoTORA simulation, control packets are required to maintain the DAG, and the size of those packets is 32 bytes. Finally, a geocast region is defined to be a 200 unit x 200 unit square region with both X and Y coordinates in the range between 500 and 700.

We use two performance metrics to measure the *accuracy* and *overhead* of geocast delivery.

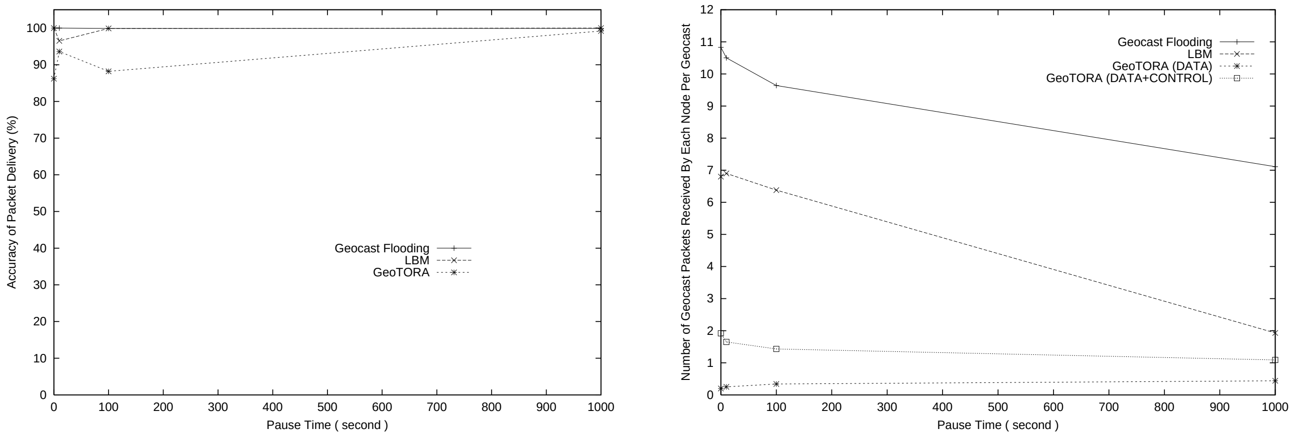
[5] [17] presents two LBM algorithms. We compare with their first algorithm which is based on flooding in a small region.

- Accuracy of Geocast Delivery [17]: *Accuracy of geocast delivery* is defined as the ratio of the number of group members that actually receive the geocast packet, and the number of group members which were in the geocast region at the time when the geocast delivery was initiated. In our simulation results, we report the average accuracy over all the geocasts performed during the simulation.

- Overhead of Geocast Delivery [17]: The *overhead* is measured in terms of the number of geocast packets *received* by the nodes – the number of geocast packets received by nodes is different from number of geocast packets *sent*, because a single broadcast of a geocast data packet by some node is received by all its neighbors. Specifically, the measures of overhead we use is the *average number of packets* and *average number of bytes* received by each node per geocast. This is calculated by dividing the total number of packets or total number of bytes received by all nodes (over a simulation run) by the number of geocasts performed, and also by the number of nodes in the system. In the pure geocast flooding and LBM scheme, the overhead is due to only data packet, but in GeoTORA it can be due to data as well as control packets.

6.2 Simulation Results

In each graph below, one parameter (e.g., pause time, maximum speed, or geocast frequency) was varied while the other parameters were kept constant.



(a) Accuracy of Geocast Delivery

(b) Total Number of Geocast Packets Received Per Geocast

Figure 8: *Comparison of GeoTORA to geocast flooding and LBM protocols with a variation of* pause time *(For 30 nodes, and Maximum speed 5.0 units/sec) : (a) Delivery accuracy versus Pause Time, (b) Total number of packets received per geocast versus Pause Time*

Figure 8(a) shows the accuracy of geocast delivery of the three geocasting protocols as a function of pause time. As can be expected, pure geocast flooding performs very well, delivering nearly 100 % accuracy. Accuracy of GeoTORA is also high, but not as high as pure flooding. As noted before, local flooding in GeoTORA may not deliver packets to all nodes in the group. This problem can be solved choosing a larger local flooding region, trading overhead with accuracy.

Delays required to establish a route to the geocast group can be another reason why GeoTORA has lower accuracy than pure flooding. Let us consider the case when the geocast region is empty. With GeoTORA, a source will not send geocast packets until it makes sure a route becomes available. In the meantime, the source will just drop packets. In contrast, both geocast flooding and LBM protocols allow the source to transmit packets in the same situation, resulting in a higher probability of packet reception by a node just entering the geocast region which was empty.

Figure 8(b) shows the overhead, i.e., average number of geocast packets received by a node per geocast, as a function of pause time. The overhead of GeoTORA consists of data packets as well as control packets (QRY, UPD, and CLR) used to create and maintain routes. The overhead due only to data packets, and overhead due to data and control packets both are plotted separately in the figure.

Generally, the overhead increases with increasing node mobility (i.e., decreasing pause time) for all schemes. However, note that the main reason for increasing overhead in GeoTORA is the control packets, not the data packets. With low mobility rate in GeoTORA, routes for forwarding packets are likely to be fixed and, therefore, the number of control packets to maintain the routes is relatively small. As mobility rate goes up, the cost for a route maintenance process, i.e., number of QRY and UPD packets, also becomes higher.

In Figure 8(b), the overhead is consistently lower for GeoTORA as compared to geocast flooding and LBM. Recall that GeoTORA limits the scope of flooding to the nodes located in the geocast region. Thus, degree of flooding is smaller in GeoTORA, compared to other two flooding-based protocols. This results in the lower overhead of GeoTORA protocol.

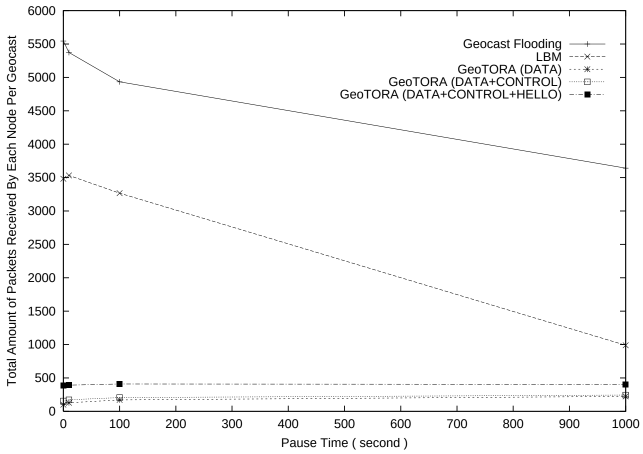
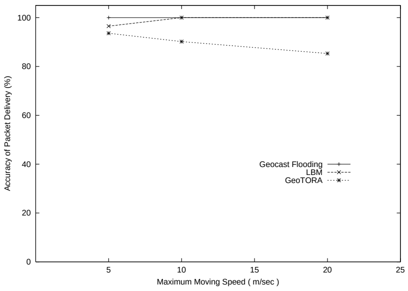


Figure 9: Total Amount of Geocast Packet Delivery (Bytes) versus Pause Time

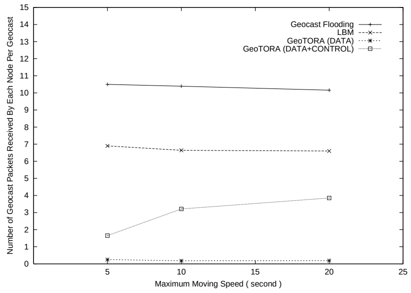
Figure 9 plots total amount (i.e., Bytes) received by each node per geocast, with results being similar to Figure 8(b). Since GeoTORA is based on TORA routing protocol layered on top of IMEP (Internet Manet Encapsulation Protocol) [22], it also uses a neighbor discovery mechanism, which requires each node to transmit at least one *hello* packet per *beacon* period (1 second). This overhead of hello packet transmission is constant and is taken into account separately from that of GeoTORA control packets, as shown in Figure 9. Observe that the geocast overhead is quite low and competitive compared to other protocols, even in the case when all different types of packets – data, control, and hello – are considered as geocast delivery overhead in GeoTORA. Note that the hello packets are also useful for unicast routing and for any other protocol that might need to detect link failures in the ad hoc networks. Therefore, It is not necessarily fair to attribute hello packet overhead to GeoTORA. However, we do so here to provide a conservative upper bound on GeoTORA overhead.

The effect of varying the moving speed of nodes is shown in Figure 10 in terms of the accuracy and delivery overhead, respectively. Increasing moving speed does not seem to have much impact on the delivery accuracy and overhead of geocast algorithms. In Figure 10(a), geocast flooding provides the highest accuracy, whereas our GeoTORA shows a slightly lower accuracy than flooding due to reasons discussed previously. However, note that geocast flooding and LBM schemes suffer from a significantly higher overhead (measured in average number of messages per geocast) than GeoTORA for all moving speed (See Figure 10(b)). Overhead measured as average bytes per geocast (as a function of speed) is also provided in Figure 10(c). We can see that GeoTORA performs much better than others even when hello packets are included in calculating the overhead.

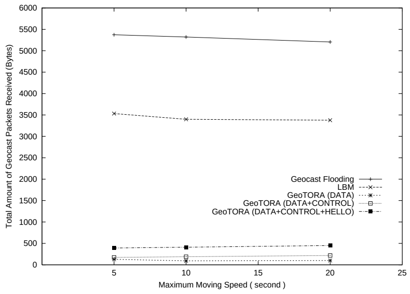
Finally, in Figure 11(a), (b) and (c), we plot accuracy and overhead (number of messages and bytes) of geocast packet delivery with varying geocast frequency. In Figure 11(b), for GeoTORA, the overhead due to data packets is almost constant. However, as pointed out earlier, GeoTORA's total overhead is due to control packet (QRY, UPD, CLR) and data packets. When geocasts are performed very infrequently, the control overhead of maintaining the DAG becomes high, therefore GeoTORA overhead becomes poor for low geocast frequency.



(a) Accuracy of Geocast Delivery

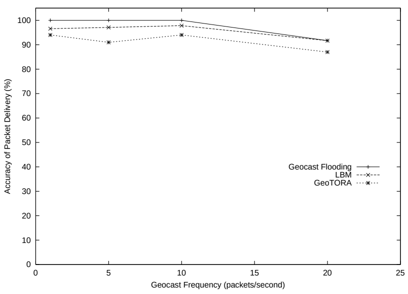


(b) Total Number of Geocast Packets Received Per Geocast

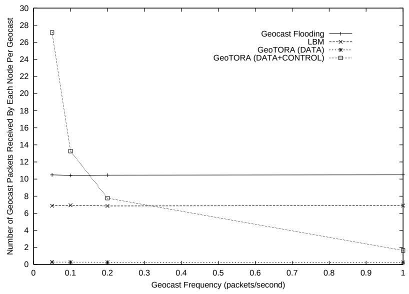


(c) Total Amount of Geocast Packets Received Per Geocast

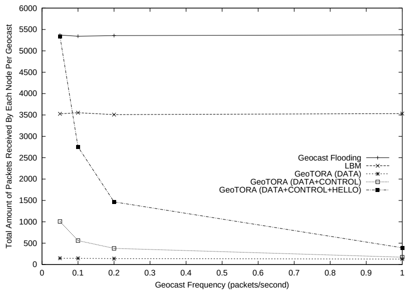
Figure 10: *Comparison of GeoTORA to geocast flooding and LBM protocol with a variation of* speed *(For 30 nodes, and Pause Time 10 seconds) : (a) Delivery accuracy versus Moving Speed, (b) Total number of packets received per geocast versus Moving Speed, (c) Total amount of packets received per geocast versus Moving Speed*



(a) Accuracy of Geocast Delivery



(b) Total Number of Geocast Packets Received Per Geocast



(c) Total Amount of Geocast Packet Received Per Geocast

Figure 11: *Comparison of GeoTORA to geocast flooding and LBM protocols with a variation of* geocast frequency *(For 30 nodes, and Pause Time 10 seconds) : (a) Delivery accuracy versus Geocast Frequency, (b) Total number of packets received per geocast versus Geocast Frequency, (c) Total amount of packets received per geocast versus Geocast Frequency*

Now observe Figure 11(c) that the overhead of data and control packets measured in *bytes* does not exceed that of LBM for the geocast frequencies simulated. This is unlike Figure 11(b), where the number of data and control packet does exceed that of LBM. The related overhead is different in the two cases because the size of control packets is much smaller than data packets. Also note that if the overhead of hello packets is charged to GeoTORA then its overhead becomes higher than LBM at low geocast frequencies.

7 Conclusion

We present a novel protocol called GeoTORA for geocasting in mobile ad hoc networks. The basic idea behind GeoTORA is to combine *anycast* and *flooding*. In GeoTORA, TORA (unicast) routing protocol has been modified to perform anycast and local flooding has been utilized to limit flood to a small region. As simulation results show, this integration of TORA and flooding can significantly reduce the geocast message overhead as compared to pure flooding and LBM scheme presented in [17], while achieving high accuracy of geocast delivery. However, note that if geocasts are performed very infrequently then the overhead of GeoTORA can exceed that of LBM.

References

[1] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, "A distance routing effect algorithm for mobility (DREAM)," in *ACM/IEEE Int. Conf. on Mobile Computing and Networking (MobiCom'98)*, October 1998.

[2] M. Bergamo, R. R. Hain, K. Kasera, D. Li, R. Ramanathan, and M. Steenstrup, "System design specification for mobile multimedia wireless networks (mmwn) (draft)," tech. rep., BBN Systems and Technologies, Oct. 1996.

[3] Bommaiah, McAuley, Talpade, and Liu, "AMRoute: Adhoc multicast routing protocol (Internet-Draft)," Aug. 1998.

[4] 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 *ACM/IEEE Int. Conf. on Mobile Computing and Networking (MobiCom'98)*, Oct. 1998.

[5] C. Chiang and M. Gerla, "Routing and multicast in multihop, mobile wireless networks," in *Proc. of ICUPC '97*, 1997.

[6] C. Chiang, M. Gerla, and L. Zhang, "Adaptive shared tree multicast in mobile wireless networks," in *Proc. of IEEE GLOBECOM'98*, 1998.

[7] C. Chiang, M. Gerla, and L. Zhang, "Forwarding group multicast protocol (FGMP) for multihop, mobile wireless networks," *ACM/Baltzer Journal of Cluster Computing: Special Issue on Mobile Computing*, vol. 1, no. 2, pp. 187–196, 1998.

[8] M. S. Corson and A. Ephremides, "A distributed routing algorithm for mobile wireless networks," *ACM J. Wireless Networks*, vol. 1, no. 1, pp. 61–81, 1995.

[9] S. Corson and J. Macker, "Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations (Internet-Draft)," *Mobile Ad-hoc Network (MANET) Working Group, IETF*, Oct. 1998.

[10] E. Gafni and D. Bertsekas, "Distributed algorithms for generating loop-free routes in networks with frequently changing topology," *IEEE Transactions on Communications*, vol. C-29, no. 1, pp. 11–18, 1981.

[11] J. J. Garcia-Luna-Aceves and E. L. Madruga, "A multicast routing protocols for ad-hoc networks," in *INFOCOM'99*, Mar. 1999.

[12] Z. J. Haas and M. R. Pearlman, "The zone routing protocol (ZRP) for ad hoc networks (Internet-Draft)," *Mobile Ad-hoc Network (MANET) Working Group, IETF*, Aug. 1998.

[13] C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath, "Flooding for reliable multicast in multi-hop ad hoc networks," in *The 3rd Intl. workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M'99)*, Aug. 1999.

[14] P. Jacquet, P. Muhlethaler, and A. Qayyum, "Optimized link state routing protocol (Internet-Draft)," *Mobile Ad-hoc Network (MANET) Working Group, IETF*, Feb. 2000.

[15] L. Ji and M. S. Corson, "LAM: Lightweight adaptive multicast protocol (Internet-Draft)," Aug. 1998.

[16] D. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing* (T. Imielinski and H. Korth, eds.), Kluwere Academic Publishers, 1996.

[17] Y.-B. Ko and N. H. Vaidya, "Geocasting in mobile ad hoc networks: Location-based multicast algorithms," in *Proc. of the 2nd IEEE Workshop on Mobile Comp. Sys. and Applications (WMCSA'99)*, Feb 1999.

[18] Y.-B. Ko and N. H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," in *ACM/IEEE the 4th Annual Intl. Conference on Mobile Computing and Networking (MobiCom'98)*, October 1998.

[19] S.-J. Lee, M. Gerla, and C.-C. Chiang, "On-demand multicast routing protocol," in *Proc. of the Wireless Communications and Networking (WCNC'99)*, Sept. 1999.

[20] J. C. Navas and T. Imielinski, "Geocast - geographic addressing and routing," in *ACM/IEEE MobiCom'97*, September 1997.

[21] K. Obraczka and G. Tsudik, "Multicast routing issues in ad hoc networks," in *IEEE ICUPC*, Oct. 1998.

[22] V. D. Park and M. S. Corson, "An internet manet encapsulation protocol (IMEP) specification (Internet-Draft)," *Mobile Ad-hoc Network (MANET) Working Group, IETF*, Nov. 1997.

[23] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *IEEE INFOCOM'97*, pp. 1405–1413, 1997.

[24] C. E. Perkins and E. M. Royer, "Ad hoc on demand distance vector (AODV) routing (Internet-Draft)," *Mobile Ad-hoc Network (MANET) Working Group, IETF*, Aug. 1998.

[25] E. M. Royer and C. E. Perkins, "Multicast operation of the ad-hoc on-demand distance vector routing protocol," in *INFOCOM'99*, Mar. 1999.

[26] R. Sivakumar, P. Sinha, and V. Bharghavan, "Core extraction distributed ad hoc routing (CEDAR)," in *Proc. of INFOCOM'99*, March 1999.

[27] C.-K. Toh, "A novel distributed routing protocol to support ad-hoc mobile computing," *Wireless Personal Communication*, Jan. 1997.

[28] C. Wu, Y. Tay, and C.-K. Toh, "Ad hoc multicast routing protocol utilizing increasing id-numbers (AMRIS) functional specification (Internet-Draft)," November 1998.