# A Geometric Routing Protocol in Disruption Tolerant Network

Jingzhe Du, Evangelos Kranakis
School of Computer Science
Carleton University
Ottawa, Canada
{jdu3, kranakis}@scs.carleton.ca

Amiya Nayak
School of Information Technology and Engineering
University of Ottawa
Ottawa, Canada
anayak@site.uottawa.ca

## Abstract

*We describe a novel Geometric Localized Routing (GLR) protocol in Disruption (Delay) Tolerant Network (DTN). Although DTNs do not guarantee the connectivity of the network all the time, geometric location information still could be used to make routing decisions in a store and forward way. Geometric planar spanners, especially local Delaunay triangulation can also be used in DTN to provide a good routing graph with constant stretch factor and shorter paths during communication. In this work, we design local distributed solutions to extract spanning trees from Local Delaunay Triangulation Graphs in the direction from source to destination. Our protocol resorts to flooding packets along the trees and with high probability packets are delivered with low delay. Through experimentation, we have shown that the proposed routing protocol achieves higher delivery ratio with lower delay and limited storage requirement than the benchmark epidemic routing protocol.*

**Keywords:** Routing, Disruption Tolerant Network, Delaunay triangulation, Spanner, Local algorithm.

**Technical areas:** Routing in ad hoc networks, Topology construction and maintenance.

## 1. Introduction

Mobile Ad Hoc Networks (MANET) consist of autonomous mobile nodes connected by wireless channels without any preexisting network infrastructure. Typically, some of these mobile devices are part of the network only while they can communicate with the rest of the network. Existing ad hoc routing protocols usually assume that there is always a connected path from message source to destination. In situations where network partitions exist, these routing protocols drop the message if a path could not be found and thus perform insufficiently in terms of message delivery. Disruption (Delay) Tolerant Networks (DTN) are proposed to address such issues in MANET where instantaneous source and destination node connections may not exist.

Geographic routing has been heavily studied in MANET. Nodes could get their location information either by global positioning system (GPS) or localization algorithm [17]. In geographic routing, a node makes routing decisions according to its neighboring nodes' location information. Since contemporaneous source to destination node connection may not exist in DTN, location inaccuracy and network disruptions have to be properly dealt with if geographic routing is applied on the network. And since different DTN networks have different network characteristics, such as different node densities and node communication ranges, a uniform routing approach is not the best choice in dealing with different situations.

There are two main categories of DTN routing protocols. Some [2, 6, 10, 14, 15, 18] are designed for specific purposes or scenarios and are not suitable for general use. For those routing protocols aiming at general scenarios, they either assume unlimited bandwidth and storage space in order to achieve desirable delay and delivery ratio attributes, or use limited resources in the cost of long delivery delay. Neither one could be easily used in DTN applications which require less delay, where nodes are normally randomly deployed, with limited storage space and bandwidth availability.

Epidemic routing [21] is a simple solution designed for general purpose use. It relies on carriers of messages coming into contact with other nodes through node mobility. When two nodes come into communication range with each other, they first exchange the information (called summary vector) which indicates the messages they hold. Messages that the other node does not have are exchanged following that. This approach can achieve high delivery ratio. If it is provided with infinite bandwidth and buffer resources, it will deliver all the messages that can possibly be delivered in the minimum amount of time without prior knowledge about the network. And because of this, it is considered as a benchmark and “unbeatable” [16].

One apparent drawback of this routing protocol lies in that the messages are never cleared. To clear the messages which have already been delivered to the destinations, some kind of acknowledgement has to be developed. K. Harras and K. Almeroth in their paper [11] present several approaches to solve this issue. Active receipt or passive receipt is generated to clear the already delivered messages. In the active mode, nodes send active receipts to inform all the nodes they meet that some messages have already been delivered to the destination. While in the passive mode, nodes only send receipts when there are some other nodes trying to send messages to them which are known to have been delivered already. But no matter what the situation is, more messages are generated in the network and how to stop the broadcasting of the receipt messages is another question.

Various protocols [4, 5, 19, 20] have been proposed to improve the efficiency of epidemic routing in recent years. While storage and bandwidth requirements are reduced in these protocols, many copies of each message are still transmitted inside the network which are never cleared.

In this paper, we propose a novel Geometric Localized Routing (GLR) scheme. It uses the localized Delaunay triangulation technique to construct a spanner. Spanning trees are then extracted from this geometric spanner, and a message is transmitted along these different trees to reduce delay in the network with intermittent connections. A node turns into store state when the message could not be delivered because of network disruption, and the delivery process is restarted after specified delay. Face routing [3, 7] technique is applied when nodes enter local minimum. In our approach, a message is transmitted in the direction from source to the destination along different routing paths to reduce delay, and message copies are controlled with intelligence to reduce resource consumption. By introducing active participation of nodes, the proposed routing protocol is closer to a practical solution than other existing ones.

We present the formal algorithm and compare it with the epidemic routing [21] and show that a) it is fast, b) it uses less storage space, c) it achieves lower delivery delay, and d) the delivery ratio is higher under limited storage space. Thus, our framework is more robust than the benchmark DTN paper [16].

The rest of the paper is organized as follows. Section 2 elaborates on our proposed solutions. Section 3 talks about the details of experiments and analysis. Section 4 concludes with possible future work.

# 2 The Proposed GLR Routing Algorithm

Our proposed solutions use geometric information to construct localized Delaunay triangulation [12]. A store and forward mechanism is added according to the special characteristics of the DTN. Controlled multi copies of same message are transmitted in the network to reduce delay. The routing protocol adjusts the number of message copies according to the connectivity characteristic intelligently.

The following goals are kept in mind in designing the proposed solution: *low delay*, *high delivery ratio* and *limited memory requirement*.

Compared with the epidemic routing, GLR achieves better storage utilization. It is faster because contentions are avoided by allowing only reasonable number of identical message copies in transit. Under limited storage restrictions, the delivery ratio of GLR is also higher.

## 2.1 Preliminaries and Notations

Our algorithm is designed to be local (for some small $k$) because it needs to make decisions that do not affect or affected by distant nodes. An algorithm is $k$-local if no message transmitted ever needs to propagate in the network more than $k$ hops.

In [13], Li *et al.* has proposed a local algorithm. Their construction is based on having each node construct the Delaunay triangulation of its distance $k$ neighborhood. Essentially, two nodes $u$ and $v$ are adjacent in their spanner iff the link $uv$ is in the local Delaunay triangulation of both $u$ and $v$. Let $N_k(u)$ be the neighborhood of node $u$.

Let $A(N)$ be the Delaunay triangulation of network $N$. In our construction, a link $uv$ is accepted in the final graph if it is in both $A(N_k(u))$ and $A(N_k(w))$, $\forall w \in N_1(u), u \in N_k(w)$ and $v \in N_k(w)$. We do this to obtain a planar graph directly, avoiding the extra time incurred by the planar process as shown in [13].

The $k$-Local Delaunay Triangulation Graph ($k$-LDTG) could be obtained by using $k$-local algorithm of Delaunay triangulation construction. For simplicity, LDTG is used in the paper to represent $k$-LDTG. The LDTG povides a good routing graph because it is a planar spanner which could be used for face routing when nodes enter local minimum. In [8], Gao *et al.* has proposed Restricted Delaunay Graph (RDG). If one only considers clusterheads and gateway nodes, 1-LDTG is same as the RDG.

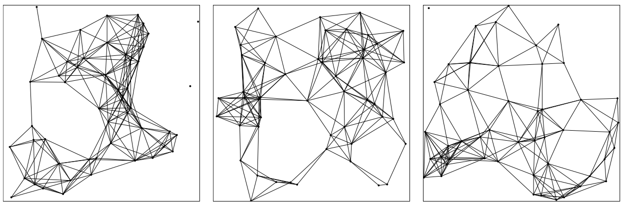
## 2.2 Delay-Tolerant Store-and-Forward

In GLR routing, flooding is controlled. When a source node has message for a destination node, it decides the number of duplicate messages required. The sparser the network is, the more copies of the same message are transmitted. Any node can calculate the network connectivity and the node density by using the number of nodes, the node communication range and the area of a given region. So the node knows the possibility of connection. In [9], Georgiou *et al.* show that for any positive real number $s$, the network $G(P, r_n)$ with a set $P$ of $n$ nodes and radius $r_n$ is connected with probability of at least $1 - \frac{1}{s}$, for $r_n \geq \sqrt{\frac{\ln n + \ln s}{n\pi}}$. The larger the node communication range, the more likely the network could be connected. If the network is dense and it could be connected at some time, single copy is enough for a fast delivery. Multiple message copies should be avoided. Otherwise, large number of contentions will lead to long delay. If it is impossible for a network to be connected, multiple copies approach should be used.

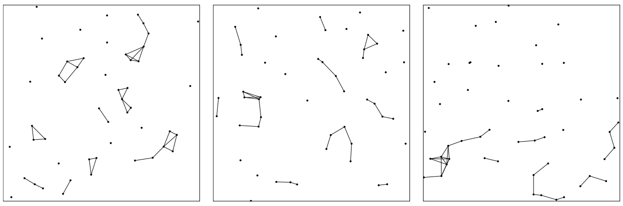
We generate scenarios with radius 250m and 100m and calculate the connection edges of 50 randomly generated nodes within area 1000m$\times$1000m. Only the edges less than the radius are kept which represent connectivity. The final results are shown in Figure 1.

These figures clearly show that when the radius is 250m, the networks are either connected or only a few nodes are disconnected with other nodes. The possibility that a source node could send a message directly to a destination node is high. Even when at the destination node could not be reached directly, with nodes moving, it is very likely that a new path will emerge. The node that holds the message waits and resends it when network topology changes with node movement. Single message copy is likely to reach the destination quickly in most scenarios. In scenario (b), the possibility of network connection is almost impossible. Compared with scenario (a), more copies of same message should be sent to increase the delivery probability and decrease delivery latency. The intelligent decision process is shown in Algorithm 1.

The decision on how many copies of a message need to be sent in Algorithm 1 above depends on network sparsity and memory storage at each sensor node in order to increase deliv-



(a) radius 250m



(b) radius 100m

Figure 1: Topology of 50 nodes with radius 250m and 100m in 1000m×1000m area.

**Algorithm 1** Delay-Tolerant Decision Making

```
procedure DELAY-TOLERANT DECISION MAKING
    if Network is sparse then
        Decide the number of message copies needed
        Send multiple copies of same message into network
    else
        Use single copy
    end if
end procedure
```

ery success and avoid contention. Details and further discussions can also be found in Subsections 3.3, 3.4 and 3.6.

### 2.3 Geometric Routing with Controlled flooding

We construct local Delaunay triangulation graph by using the $k$-local algorithm. Before we route the packet in a greedy manner to the next node closer to the destination, three trees are extracted from the underlying geometric spanner using methods similar to that are described in [9]. The difference lies in that the direction of the tree extraction is from source to the destination. We call these three trees the *max distance source to destination tree* (MaxDSTD), the *min distance source to destination tree* (MinDSTD), and the *mid distance source to destination tree* (MidDSTD). In MaxDSTD, each node is connected to a neighbor that makes maximum progress (e.g., closest) to the destination while in MinDSTD and MidDSTD, each node is connected to a neighbor that makes minimum and medium (between maximum and minimum if it is possible) progress to the destination. By using the extracted trees, a message travels from source to the direction of destination along different paths. Compared with only one routing path, these trees provide faster delivery in a sparse networks even when there are disruptions. Unlike MaxDSTD and MinDSTD trees, the MidDSTD tree has more options, i.e., a node may have several mid distance neighbors which can make progress to the destination and any one could be selected in the tree. If more than three identical message copies are needed by a source node's intelligence decision process, multiple MidDSTD trees are extracted.

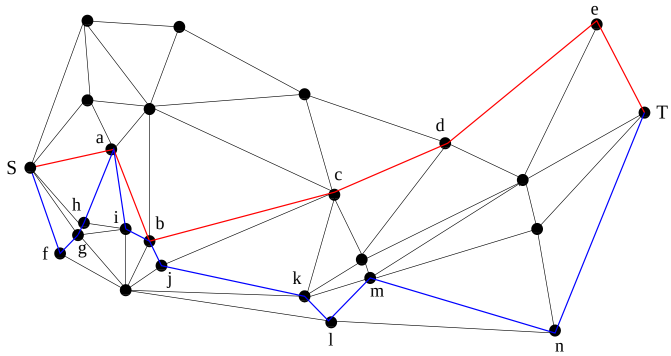



Figure 2: MaxDSTD and MinDSTD trees

Figure 2 is an illustration of extracting MaxDSTD and MinDSTD trees. When a source node $S$ has message for destination node $T$, it decides the number of message copies first. If multi-copies approach is adopted, multiple trees are extracted along the path in the direction from source to the destination. In MaxDSTD tree, a message follows the route $S \rightarrow a \rightarrow b \rightarrow c \rightarrow d \rightarrow e \rightarrow T$. While in MinDSTD tree, the route is $S \rightarrow f \rightarrow g \rightarrow h \rightarrow a \rightarrow i \rightarrow b \rightarrow j \rightarrow k \rightarrow l \rightarrow m \rightarrow n \rightarrow T$. These two trees are different and message follows different routing paths. With nodes moving, more routing paths increase the chance of delivery and reduce delay. Our geometric routing process is shown in Algorithm 2.

In our algorithm, *Messagecount* is the number of messages a node has. A node initiates the geometric routing process if it has messages in its storage area. The LDTG is constructed first. Messages are treated differently if their corresponding destinations are different. If a node finds closer neighbors for a message, it removes the message from message storage area and decreases the *Messagecount*. The source decides the number of identical message copies needed, extracts corresponding next hop from the LDTG (only MaxDSTD if single copy approach is chosen) and sets flags for the messages. A relay node only needs to route message following the tree specified by the message flag.

When a node comes to a state when no action could be taken for a message, it stores the message and forwards it later. A timer is used when a node enters into store state. When the timer expires (the interval is *checkinterval*), the node checks neighboring nodes' locations to see if any change has occurred. When its relative location with respect to the neighboring nodes changes and new path emerges in the locally constructed trees, it will send the stored messages.

#### 2.3.1 Location Diffusion

For the geometric routing to function properly, the destination node's location accuracy is required in making routing decisions. Although it is assumed in the GLR that source node knows the destination location information at the beginning, it is possible that the destination node has moved far away from its initial location during the process of message delivery, especially in a network with long delays and disruptions.

To address this issue, location information diffusion is ap-

**Algorithm 2** Geometric Routing with Controlled Flooding

**procedure** GREEDYGEOMETRICROUTING
  **while** $Messagecount \neq 0$ **do**
    *Collect neighboring nodes information*
    Construct LDTG
    **for** Every Message **do**
      **if** There are neighbors closer to destination **then**
        $Messagecount \leftarrow Messagecount - 1$
        Release message storage space
        **if** Source node **then**
          **if** Need more copies **then**
            Extract Max(Min,Mid)DSTD neighbors
            Message $max(min, mid)flag \leftarrow 1$
            Send message to these neighbors
          **else**
            Extract MaxDSTD neighbor
            $maxflag \leftarrow 1$
            Send message to MaxDSTD neighbor
          **end if**
        **else**
          **if** Message maxflag == 1 **then**
            Extract MaxDSTD neighbor
            Send message to MaxDSTD neighbor
          **end if**
          **if** Message minflag == 1 **then**
            Extract MinDSTD neighbor
            Send message to MinDSTD neighbor
          **end if**
          **if** Message midflag == 1 **then**
            Extract MidDSTD neighbor
            Send message to MidDSTD neighbor
          **end if**
        **end if**
      **end if**
    **end for**
    $Wait(checkinterval)$
  **end while**
**end procedure**

plied in the process of message exchange. Two nodes exchange their location information whenever they come within communication range of each other. The location information is recorded, together with the time stamp. Each node keeps a table of other nodes' location information together with their IDs and time stamps. Message holder adds destination location information in the packet which is used to collect neighboring nodes' information. A neighboring node updates the destination location information if the message holder has more recent destination location than its own and notifies the message holder if it has more recent destination location than that of the message holder.

For best location accuracy, location tables should be exchanged whenever two nodes meet each other. Since this will add extra overhead in the routing protocol, it is not used in the experimentation of GLR considering the above mechanism is enough for the geometric routing to function properly.

#### 2.3.2 Custody Transfer

In the proposed routing solution, custody transfer is used to ensure that a message is not be deleted by the sender unless the corresponding receiver has notified the sender that it has received the message. Two storage areas are maintained to distinguish between the messages which have not been sent and the messages which have been sent and waiting to be acknowledged. The *Store* is the place where messages are waiting to be sent whereas messages that are just sent are saved in the *Cache*.

Whenever a node successfully receives a message, it notifies the sender that the message has been received correctly. This notification contains information regarding the source node, destination node, message count (the $i^{th}$ message generated by the source node) and the extracted tree branch information (it is needed because messages in different tree branches follow different routing paths). After receiving reply from the receiver, the sender checks the *Cache* and deletes the corresponding message from it.

In the case that a message was lost during transfer or reply was not received properly, after staying in the *Cache* for specified time, the message is moved from *Cache* to *Store* for another round of transfer rescheduling and may or may not choose the same next hop this time when position, neighboring nodes and destination location have changed.

## 3 Experimental Evaluation

In order to evaluate our geometric routing strategy, we perform simulations to compare GLR with epidemic routing. During the experiments, we pay great attention to the key routing attributes, including message delivery latency, delivery ratio, and storage usage. Data concerning other routing parameters is also collected to present facets of the proposed solution. Simulation results show that GLR is faster than epidemic routing, with higher delivery ratio and better storage utilization.

### 3.1 Simulation Environment

The GLR is implemented using the NS-2 [1] simulator. This simulation environment includes full simulation of the IEEE 802.11 physical and MAC layers, which makes the simulation better reflect the real world. Random waypoint model is chosen as the motion pattern. For the propagation model, we have chosen *Two Ray Ground* which considers both the direct path and a ground reflection path. Nodes collect distance two neighborhood information to construct LDTG in the experiments. The simulation parameters are shown in Table 1.

Table 1: Parameters of the simulations.

| Parameter | Value |
|---|---|
| Number of mobile nodes | 50 |
| Mobility | 0-20m/s(uniform distribution) |
| Transmission range | 50-250m |
| Data rate | 1 Mbps |
| Propagation model | *Two Ray Ground* |
| Simulation time | 1200/3800(default) seconds |
| Link layer queue length | 150 |
| Topology size | 1500m $\times$300m |
| Pause time | 0 seconds |
| Packet payload size | 1000 bytes |
| Antenna model | OmniAntenna |

The GLR is layered on top of Internet MANET Encapsulation Protocol (IMEP). By modifying the IMEP packet header format in NS-2, each node adds its location information to the header and exchanges this information with its neighbors in the process of IMEP Link/Connection Status Sensing. Location information provided by IMEP is used in the location diffusion process. Since the IMEP layer updates neighbor information at specified time interval, the location information is not accurate. Because of this, a node also acquires location information from its neighbors during the routing control message and data exchange process in the implementation.

Various scenarios are simulated during experimentation. When comparing with epidemic routing, we use the same number of messages as described in [21]. A subset of 50 nodes act as sources and destinations, with each of 45 nodes sending packets to other 44 nodes (1980 messages total). Packets are generated every second.

We make the following assumptions in the system evaluation:

- Source knows the true destination location,
- Nodes have synchronized clock.

For the simulation results, all points in the figures, as well as numbers in all the tables are obtained as an average of 10 different runs with 10 different network topologies and movement patterns. The confidence intervals for the numbers are calculated at 90% confidence level.

### 3.2 Store and Forward Mechanism

When delivery route is not available at the time of sending the message, node will store the message in its storage area, waiting for a while and trying to see if possible routes appear later. The node which has messages to send needs the destination node location as well as its neighboring nodes location information to make routing decision. Neighboring nodes location information is obtained by asking, and waiting for all the replies coming in. In this process, new messages may be received, but only could be stored at the end of the message queue, waiting for their turn to be transmitted. When there are messages in store, route availability is checked periodically. The check interval affects the delivery latency also, although not too much. The more frequently route check is performed, the more control messages are in transit, but delivery latency is reduced in general. Figure 3 shows this tradeoff. Simulations are performed with 1980 messages in 100m scenario.

Our simulation uses 0.9 seconds as the default check interval.

### 3.3 Location update

In geometric routing, destination node's location information is important for nodes in the routing path to make routing decisions. Delivery latency differs greatly when the location availability is different, especially in the DTNs when destination nodes could move far away from the their original places when the messages destined for them were generated. Four different situations are considered in the simulation. The first

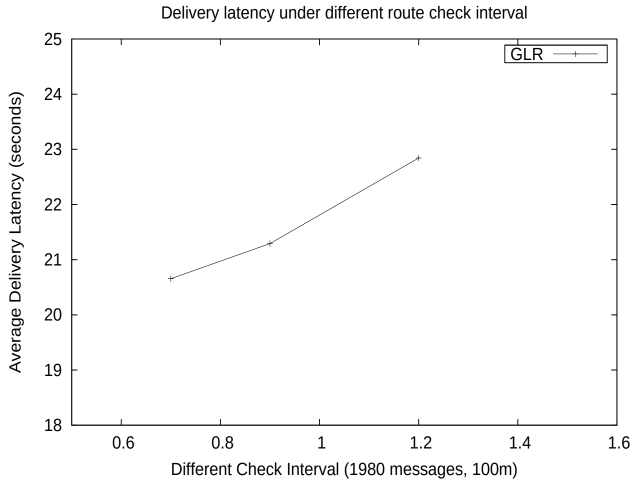



Figure 3: Delivery latency comparison (different route check interval)

one assumes all nodes in the path from source to the destination node know exactly the destination location. The second and third assume only the source node knows the destination node location and includes the $x$ and $y$ coordinates of the destination node in the messages. The last situation assumes no node knows the destination location information well in advance. The simulation results are presented in Table 2.

Table 2: Message delivery results under location information availability

| Number of Copy | Destination Location | Delivery Rate | Average Latency (seconds) | Average Hops | Storage (number of messages) |
|---|---|---|---|---|---|
| 1 copy | All nodes know | 100% | 120.2± 8.5 | 14.9± 0.3 | 38.3± 1.4 |
| 3 copies | Only source knows | 100% | 149.7± 9.6 | 17.3± 0.4 | 43.6± 1.4 |
| 1 copy | Only source knows | 100% | 156.1± 11.2 | 18± 0.3 | 40.3± 2 |
| 3 copies | No nodes know | 99.9% | 212.4± 16.6 | 23.1± 0.5 | 50.9± 3.8 |

Location information is exchanged locally throughout the network. Whenever two nodes come within communication range of each other, they exchange their location information. In the message delivery process, the destination location is updated when the message owner knows more recent destination location information than its own.

The above results show that three copies approach with source knows destination location is slower than the one copy approach with all nodes know destination location. But it performs better than the one copy approach with source knows destination location. This reflects the fact that the controlled flooding really reduces latency.

Although source node knows nothing about the destination node location in the last situation, relay nodes could adjust the destination location in the process of delivery (random location is given at the beginning). Because of unknown destination location, delivery latency of the last approach is the longest and not all messages are delivered to the destination nodes within specified time frame (3800 seconds in the simulation) as a result.

*The impact of location inaccuracy and solution*: In the process of delivery, the destination node could move away from the original place where it stayed when the message destined to it was generated. When the message reaches a node that is closest to a stale destination location which is contained in the packet header, no neighboring node could be selected as the next relay because this node is the closest.

To avoid long time delay, a new value is assigned to the destination location so that the node which is closest to the wrong location could deliver it out to another node to increase the delivery probability.

## 3.4 Delivery Latency Comparison

In epidemic routing, nodes flood their neighbors with all the messages they hold. This approach costs significant storage space. At first, this protocol appears to deliver messages faster than other approaches, but through simulation we show that the proposed routing protocol outperforms epidemic routing in delivery latency when messages in transit increase. Figure 4 shows the delivery latency comparison for radius of 50m.

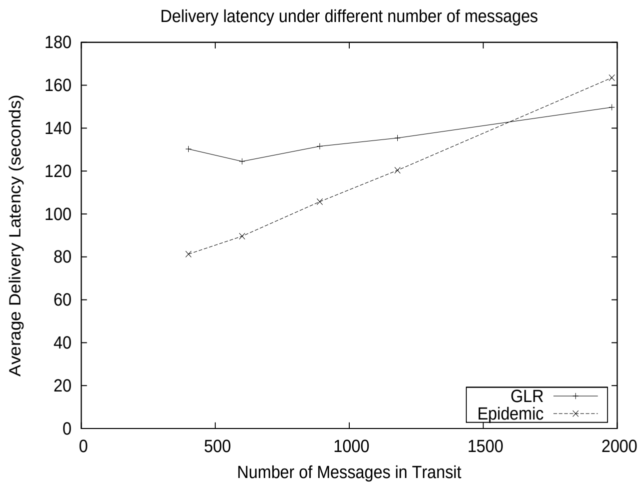



Figure 4: Delivery latency comparison (50m radius)

The increased contention is the reason why epidemic routing slows down when messages increase.

In 100m radius, the delivery latency of our routing protocol also outperforms epidemic routing. Figure 5 shows the result.

Under different radius, the delivery latency of our routing protocol also outperforms epidemic routing in 1980 messages scenario. Figure 6 shows the result. For the radius 150, 200 and 250 meters scenario, single copy approach is used. The use of more copies in these situations increases contention severely and leads to long delay.

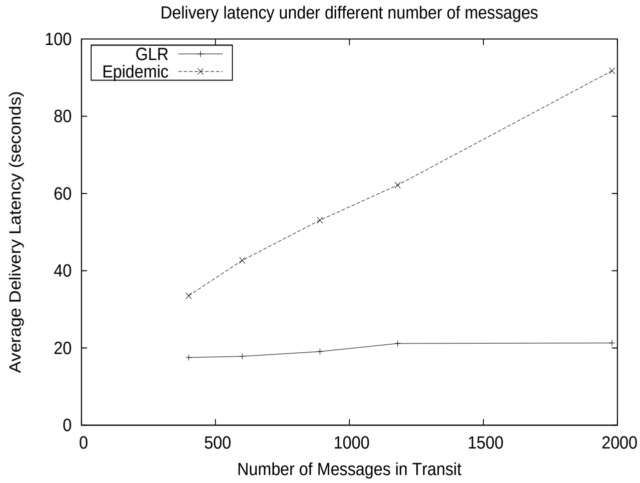



Figure 5: Delivery latency comparison (100m radius)

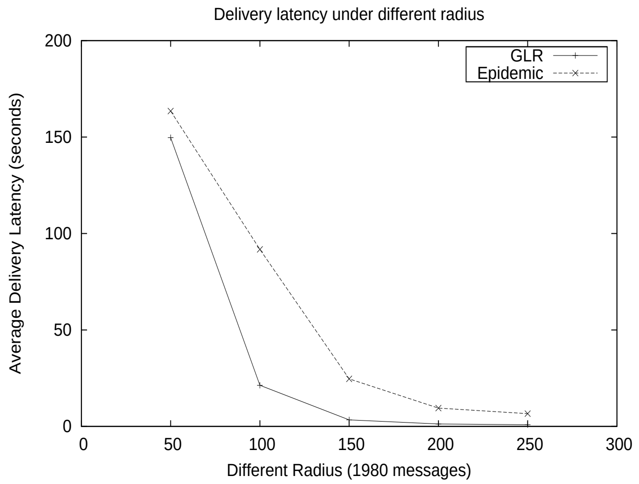



Figure 6: Delivery latency comparison (various radius)

## 3.5 Delivery with Custody Transfer

Without custody transfer, messages could be delivered with high probability but without any guarantee. With custody transfer, a message is discarded only when the sender receives an acknowledgment from the receiver. Table 3 shows the results of delivery ratio comparison. The simulation time in this case is 1200 seconds.

Table 3: Message delivery ratio comparison (50m)

| Simulation Scenario (Number of Messages) | 890 messages |
|---|---|
| Delivery ratio without custody transfer | 84.7%±1% |
| Delivery ratio with custody transfer | 97.9%±1% |

The delivery ratio will be different in different scenarios, but what appears to be clear is that hardly could all messages be delivered without custody transfer because of contention or node movement.

### 3.6 Delivery Ratio Comparison

The proposed DTN routing protocol achieves 100% delivery ratio, same as its epidemic counterpart when storage is unlimited. When storage drops below 200 messages/node in the case of 1980 messages in transit scenario, the delivery ratio of epidemic routing begins to drop, while the proposed approach still maintains 100% delivery ratio even when the storage drops to 100 messages/node.

Figure 7 gives the simulation results of the delivery ratio comparison between the proposed solution and epidemic routing.

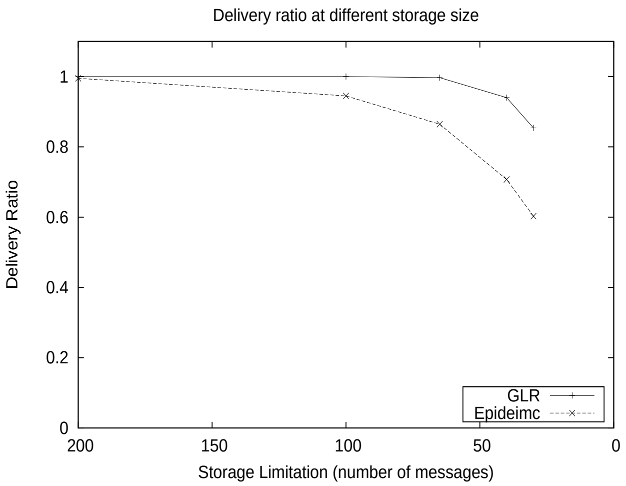



Figure 7: Delivery ratio comparison (50m radius)

In the epidemic routing, messages will never be dropped when nodes are assumed to have unlimited buffer space. The reason for keeping all messages is that nodes do not know whether a message has been delivered to the destination. When storage is limited and the storage space is fully occupied, old messages are dropped when new messages come in. FIFO queue and cache are used to handle messages. In the proposed solution, a message is deleted from the storage when it has been delivered to the next hop node and acknowledgment is received from that node. When storage space is not enough, message in the *Cache* is dropped first. Because flooding is controlled in the proposed solution, less storage space is needed in total. Under limited storage situation, less messages are dropped in our solution compared with the epidemic routing, as shown in Figure 7.

### 3.7 Storage Requirements

Storage space is required in the process of message delivery when a node is busy or delivery path to the destination is not available. When the number of messages in transit increases, the storage required also increases. It is also true that the longer the radius, the smaller is the storage requirement. This could be verified in the Tables 4 and 5.

Since the amount of storage space required in epidemic routing is same as the number of messages in transit, our routing protocol saves considerable storage space compared with its epidemic counterpart.

Table 4: Storage requirement (number of messages), different message numbers (50m, 3copies)

| Number of messages | 400 | 600 | 890 | 1180 | 1980 |
|---|---|---|---|---|---|
| max peak storage | 39± 4.67 | 43.9± 3.38 | 49.1± 2.97 | 59.9± 7.17 | 69± 5.82 |
| average peak storage | 21.31± 0.59 | 25.77± 1.05 | 30.2± 1.23 | 37.28± 2.82 | 43.64± 1.42 |

Table 5: Storage requirement (number of messages), different radius (1980 messages, 3 copies for 50m/100m and 1 copy for 150m/200m/250m)

| Simulation Scenario | 250m | 200m | 150m | 100m | 50m |
|---|---|---|---|---|---|
| max peak storage | 6.9± 4.29 | 14.3± 4.81 | 24.3± 4.54 | 48.4± 6.52 | 69± 5.82 |
| average peak storage | 1.76± 0.72 | 3.28± 1.06 | 8.36± 0.95 | 25.82± 1.37 | 43.64± 1.42 |

### 3.8 Hop Count Comparison

In the proposed routing protocol GLR, every node which has messages in its storage checks regularly to see if there are other nodes closer to the destination. In the epidemic routing, nodes exchange messages only when they come within the communication range of each other, and no message is exchanged even if relative positions have changed. As a result, a message in the case of geometric routing protocol travels along more hops than that in the case of epidemic routing. Table 6 shows the simulation results.

Table 6: Hop counts (1980 messages, for geometric routing, 3 copies for 50m/100m and 1 copy for 150m/200m/250m)

| Simulation Scenario | 250m | 200m | 150m | 100m | 50m |
|---|---|---|---|---|---|
| GLR routing | 3.4± 0.04 | 4.1± 0.05 | 5.23± 0.13 | 8.75± 0.13 | 17.32± 0.4 |
| Epidemic routing | 3.19± 0.14 | 3.64± 0.07 | 4.58± 0.07 | 4.92± 0.06 | 3.92± 0.05 |

## 4 Conclusions

We have proposed a novel routing mechanism, called GLR, that uses local neighborhood location information to construct localized planar spanners. Source nodes control message flooding with intelligence. Geographic routing is then used to deliver message in a greedy manner to the node closer to the destination. Spanning trees are used to achieve better delay tolerance. We use store and forward technique to deliver message upon partition. Complementary techniques are employed to improve location accuracy and ensure message delivery in every step. When nodes enter local minimum, the underlying planar spanner provides better routing graph for face routing. Simulation results show that GLR outperforms epidemic routing in randomly generated networks with respect to delivery delay, storage utilization and delivery ratio under limited storage space.

## References


[1] The Network Simulator, NS-2, `http://www.isi.edu/nsnam/ns/`, Accessed July 8, 2008.

[2] C. Becker and G. Schiele, "New mechanisms for routing in ad hoc networks through world models," *Proceedings of the 4th CyberNet Plenary Workshop, Pisa, Italy*, 2001.

[3] P. Bose, P. Morin, I. Stojmenović, and J. Urrutia, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks," *Wireless Networks*, vol. 7, no. 6, pp. 609–616, 2001.

[4] B. Burns, O. Brock, and B. Levine, "MV routing and capacity building in disruption tolerant networks," *Proceedings of IEEE INFOCOM*, vol. 1, 2005.

[5] J. Davis, A. Fagg, and B. Levine, "Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks," *5th International Symposium on Wearable Computers, Zürich, Switzerland*, pp. 141–148, 2001.

[6] M. Demmer and K. Fall, "DTLSR: delay tolerant routing for developing regions," *Proceedings of Workshop on Networked Systems for Developing Regions*, 2007.

[7] H. Frey and I. Stojmenovic, "On delivery guarantees of face and combined greedy-face routing in ad hoc and sensor networks," *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking*, pp. 390–401, 2006.

[8] J. Gao, L. Guibas, J. Hershberger, L. Zhang, and A. Zhu, "Geometric spanners for routing in mobile networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 174–185, 2005.

[9] C. Georgiou, E. Kranakis, R. Marcelín-Jiménez, S. Rajsbaum, and J. Urrutia, "Distributed Dynamic Storage in Wireless Networks," *International Journal of Distributed Sensor Networks*, vol. 1, no. 3, pp. 355–371, 2005.

[10] R. Handorean, C. Gill, and G. Roman, "Accommodating transient connectivity in ad hoc and mobile settings," *Lecture Notes in Computer Science*, vol. 3001, pp. 305–322, 2004.

[11] K. Harras and K. Almeroth, "Transport layer issues in delay tolerant mobile networks," *Proceedings of IFIP-TC6 Networking*, 2006.

[12] J. Keil and C. Gutwin, "The delaunay triangulation closely approximates the complete euclidean graph," *Proc. 1st Workshop Algorithms Data Structure (LNCS 382)*, 1989.

[13] X. Li, G. Calinescu, and P. Wan, "Distributed construction of a planar spanner and routing for ad hoc wireless networks," *Proceedings of IEEE INFOCOM*, vol. 3, 2002.

[14] S. Merugu, M. Ammar, and E. Zegura, "Routing in Space and Time in Networks with Predictable Mobility," *College of Computing Technical Reports, Georgia Institute of Technology, at `http://smartech.gatech.edu/handle/1853/6492`,2004. Accessed July 25, 2008.*

[15] A. Pentland, R. Fletcher, and A. Hasson, "DakNet: Rethinking Connectivity in Developing Nations," *IEEE Computer*, vol. 37, no. 1, pp. 78–83, 2004.

[16] R. Ramanathan, P. Basu, and R. Krishnan, "Towards a formalism for routing in challenged networks," *Proceedings of 2nd Workshop on Challenged Networks*, pp. 3–10, 2007.

[17] A. Savvides, C. Han, and M. Srivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pp. 166–179, 2001.

[18] C. Shen, G. Borkar, S. Rajagopalan, and C. Jaikaeo, "Interrogation-based relay routing for ad hoc satellite networks," *Proceedings of IEEE GLOBECOM*, vol. 3, 2002.

[19] T. Small and Z. Haas, "Resource and performance tradeoffs in delay-tolerant wireless networks," *Applications, Technologies, Architectures, and Protocols for Computer Communication*, pp. 260–267, 2005, ACM Press New York, NY, USA.

[20] K. Tan, Q. Zhang, and W. Zhu, "Shortest path routing in partially connected ad hoc networks," *Proceedings of IEEE GLOBECOM*, vol. 2, 2003.

[21] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," *Duke University, at `http://www.cs.duke.edu/~vahdat/ps/epidemic.pdf`, 2000. Accessed July 25, 2008.*