Evaluation of Geocast Routing Trees
on Random and Actual Networks

Bernd Meijerink[(✉)], Mitra Baratchi, and Geert Heijenk

University of Twente, Enschede, The Netherlands
{bernd.meijerink,m.baratchi,geert.heijenk}@utwente.nl

Abstract. Efficient geocast routing schemes are needed to transmit messages to mobile networked devices in geographically scoped areas. To design an efficient geocast routing algorithm a comprehensive evaluation of different routing tree approaches is needed. In this paper, we present an analytical study addressing the efficiency of possible routing trees for geocast packets. We evaluate the Shortest Path Tree, Minimum Spanning Tree and a Steiner Heuristic based routing tree for geocast packet distribution on real world networks and random graphs. We compare the results to those for multicast routing for which such evaluations have been performed in the past. Our results show that due to the correlation of geographic distance and network distance in most wired networks, Shortest Path forwarding efficiency can come close to an ideal Steiner Tree. We also identify a correlation between the forwarding efficiency and network characteristics such as the node degree and betweenness. This information could be useful in deciding on a choice of routing method or even help with network design.

Keywords: Geocast · Multicast · Routing · Shortest Path Tree · Steiner tree

1 Introduction

With the increase in the number of networked devices in the world, be it intelligent vehicles or household appliances, new communication methods are needed to allow efficient communication to and from these devices with a certain geographical constraint such as a street or district [1]. This can be achieved through geocast, first proposed by Navas and Imielinski [2]. Geocast is the transmission of packets towards a geographical area instead of a fixed address, devices receive packets purely based on their location.

An alternative and more explored method to geocast is multicast. Both of these schemes transmit packets to multiple destinations. They also share forwarding characteristics in that packets are only duplicated when the path in the network diverges. Unlike multicast the destination of packets in geocast share a geographical region and they are not distributed throughout the network. Furthermore, unlike multicast a device cannot simply subscribe to a group to receive

a geocast packet. The geocast packet is transmitted to all devices on a network in a specific geographic region. These characteristics are especially beneficial for transmission towards vehicular networks, where nodes are mobile and keeping track of membership information and location is inefficient [1].

The routing requirements for geocast differ from multicast in several ways: (1) There is a logical correlation between the geocast address and the area a packet needs to be forwarded to. (2) Routing is based on an geocast address, not membership information.

In a multicast scenario the routers that need to be reached can be distributed throughout a network. In the geocast case, these routers would be located close to each other geographically. While geographic distance does not directly correlate to network distance, a strong link between both of them can be observed in a large number of real world networks. Our hypothesis is that this geographic clustering will lead to a situation were a geocast source has an obvious forwarding path to the destination routers. This could result in a significant portion of Shortest Path routes from the source to the destination being shared. Therefore, due to the geographically scoped nature of geocast, traditional routing methods such as unicast or multicast will not potentially provide the required efficiency. A new set of routing algorithms specifically designed for geocast is needed to provide an effective geocast solution in Internet-scale networks [1].

To design an efficient geocast routing algorithm we require information on the efficiency of different possible forwarding trees. Our hypothesis is that more optimal methods like Steiner trees are not as relevant when destinations are located close to each other and simpler but computationally less expense methods such as naive Shortest Path forwarding are more attractive. Our assumption is that routers that are responsible for areas in close geographical proximity, are also close to each other in the network with a small amount of hops between them.

The main contribution of this paper is to identify a forwarding tree that can be used for the design of an efficient geocast routing algorithm. This is done by performing an extensive evaluation of different forwarding trees in a gecoast scenario. We use the average cost and path utilization over multiple (source, destination) pairs as our main metrics. We compare the results with results from multicast based evaluations. The multicast case has been extensively researched in the past [3,4], but the effect of geographical clustering on the forwarding tree efficiency is an open question. This information can be used in later work do design an efficient routing system for geocast traffic. In our previous work we proposed an addressing system for Internet wide geocast [5]. This system can address rectangular areas with a minimum size of 7 by 3.5 cm and is logically routable using a form of prefix matching. Combined with an efficient routing mechanism this could potentially allow geocast in Internet-scale networks.

This paper is structured in the following way: In Sect. 2 we explore previous work on the topic of multicast Shortest Paths and random graphs. Section 3 explains our evaluation approach and which metrics we use. Our results are described and discussed in Sect. 5. Finally we draw our conclusions and discuss future work in Sect. 6.

2   Previous Work

In our previous work on geocast addressing we proposed an addressing scheme for geocast [5]. This addressing scheme allows routing based on a type of prefix matching. To implement an efficient routing method we would need to know which type of routing tree has the best characteristics in terms of links used for geocast routing in real world networks.

Previous papers have explored the benefit of using different forwarding mechanisms for multicast traffic. The authors of [3] show that a naive Shortest Path Tree from the source is not that much more inefficient than a Steiner tree heuristic method. Their evaluation focuses on multicast performance in Waxman graphs.

In [4] the authors evaluate different multicast trees for their properties in overall cost and delay. They show that a Shortest Path Tree based approach can come close to the Steiner tree heuristics in terms of performance.

More recently the focus of this kind of evaluation has been in the realm of ad-hoc wireless networks. In [6] Nguyen et al. show that Shortest Path Trees provide benefits over minimum cost trees in wireless ad-hoc networks. According to the authors these benefits outweigh the downside of higher tree cost.

Knight et al. have published a database of public network topologies at the PoP level [7]. They perform a statistical analysis on the data and map the properties such as node degree of these network. We use the actual networks published in the Topology Zoo in our evaluation and use the statistical data to generate random geometric graphs.

Constructing a Steiner Tree over a graph is a NP-complete problem. Kou et al. have presented a fast Steiner Heuristic algorithm [8]. We use the algorithm to find the Steiner Tree for our route evaluations. This allows our evaluation to contain a larger number of graphs than would otherwise be possible. It also has the benefit of being more close to a solution that could be used in an actual router for tree construction.

3   Approach

In this section, we will explain our approach to evaluate the three routing trees. We will first present the trees with their advantages and drawbacks, followed by a short presentation of the tools and sources used. To perform a fair evaluation of the three different routing tree approaches in a geocast scenario we will use two graph models. We will generate random geometric graphs to create a set of networks on which we can perform evaluations, and we will use actual network topologies used in the real world. Information relevant to these graphs will be presented at the end of this section.

3.1   Routing Trees

We evaluate three methods of geocast and multicast trees that can be realistically used for routing: (1) Shortest Path Tree from source, (2) Minimum Spanning Tree, (3) Steiner Tree from source.

Each of these approaches have different benefits and drawbacks that will make them more or less suitable depending on the goals of the network administrator or even the layout of the network.

The Shortest Path Tree is simply a combination of all Shortest Paths from the source to the destination nodes. We count a link that is used multiple times as one usage, as we assume an underlying routing protocol can prevent duplicate packets over the same link. For example: Router A needs to forward a message to a specific area which includes router B, C and D. The Shortest Path Tree would be the union between the shortest paths $(A \rightarrow B)$, $(A \rightarrow C)$ and $(A \rightarrow D)$. Using Fig. 1a as an example: Using node 6 as the source and nodes 8, 9 and 10 as destinations the Shortest Path Tree would consist of $6 \rightarrow 7 \rightarrow 8, 6 \rightarrow 10 \rightarrow 9$ with a total cost of 4. This approach requires a per (source, destination) pair forwarding calculation for each router. A simple per destination forwarding calculation as would be the case for unicast is not possible. As it is probable that the destination area includes multiple routers, a forwarding router needs at least some knowledge of how it fits in the distribution tree to make an efficient forwarding decision. We suspect that this approach will be efficient for geocast as the geographic closeness of destination likely strongly correlates to closeness in the network leading to a large number of shared links.

For the Minimum Spanning Tree, we simply calculate the Minimum Spanning Tree of the network (based on hop count). This subgraph is used to reach all destination nodes from the source. This approach has the benefit that the distribution tree for any geocast (or multicast) message can be precomputed. The major downside is that several links will carry all the traffic, while others are never used. This approach will also not lead to the lowest overall path cost as the most efficient route will almost never be used in most networks. It can however, perform equal to the Steiner tree in situations were the source and destination nodes are ideally distributed on the Minimum Spanning Tree. However, this situation is not likely to occur often and will be offset by all the destinations that are not ideally distributed on the tree.

A Steiner tree is the least cost tree between source and destination nodes. Because this is a NP-complete problem we use a well known heuristic algorithm [8] to construct it. This algorithm works by first finding the metric closure of the
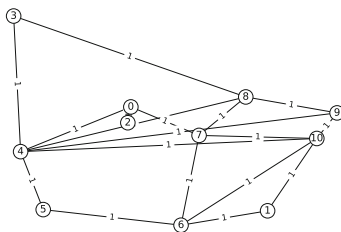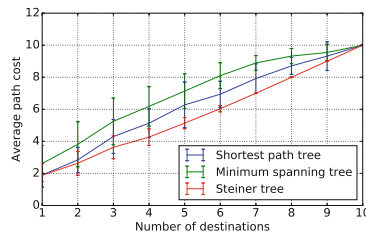
(a) Network graph   (b) Path cost example

Fig. 1. A real world network (Color figure online)

nodes we are interested in. The Minimum Spanning Tree is calculated over the metric closure graph and we map this back to the actual network. This approach will lead to a close to optimal graph but like the Shortest Path approach we need to compute a tree for each (source, destination) pair, with higher computational overhead. Again using Fig. 1a as an example, with node 6 as the source and nodes 8, 9 and 10 as destinations: The Steiner tree would consist of $6 \rightarrow 10 \rightarrow 9 \rightarrow 8$ with a total cost of 3 (one less compared to the Shortest Path Tree). As mentioned before, the Steiner tree is the least cost tree but has the downside of requiring more overhead to computer compared to the other two trees. In the geocast scenario a forwarding router would need knowledge of the source router and all destination routers to know its place in the ideal forwarding tree.

3.2 Tools and Sources

To perform our evaluation, we used several preexisting tools. To model and evaluate graphs we used the NetworkX package [9] for the python programming language. The random graphs used were also generated using this package. All the real world graphs we evaluated are taken from the Topology Zoo [7].

3.3 Networks

For the rest of this paper, we will refer to a graph $G = (V, E)$, with V the vertices or nodes (representing routers), E the edges (representing links). We use both real word networks and randomly generated graphs in our evaluation.

Real Networks. To perform a fair evaluation of the different approaches we need to consider real world networks, both as a control sample and a validation of the random geometric graphs. A computer network is by definition a designed system that is built in a certain way for specific reasons such as cost, performance or necessity. This also means that nodes close to each other are not always connected due to reasons such as geography or politics that we cannot easily fit into a graph.

We use several network graphs that have been made available through the Topology Zoo project [7]. We import these graphs and remove all nodes that are not connected to other nodes. When the resulting graph is still disconnected, we take the largest subgraph as the graph to run our evaluation on. In the majority of cases the graphs can be imported without these operations. One example of such a graph is the one depicted in Fig. 1a. This graph will be used later to explain our evaluation process.

Random Geometric Graphs. To supplement the actual networks used and provide a basis for more general conclusions we have also generated a set of random geometric graphs to run our evaluation on. We chose to use random geometric graphs because the presence of edges between vertices is based on geometric distance. This property is helpful in geocast evaluation as it provides a strong correlation between network distance and the relative distance between

nodes. We acknowledge that a random geometric graph may not represent an actual network with high accuracy, but the set of actual networks should sufficiently cover this, allowing the random networks to focus on an ideal geocast case.

The graphs were generated using varying numbers of nodes and accepted as valid based on three criteria:

1. The graph is connected (every node is reachable by every other node).
2. The average betweenness centrality is similar to the studied real graphs (between 0 and 0.3). The betweenness centrality is a measure of the importance of a node, it is the fraction of shortest paths between node pairs that pass through it [10]. The average gives an indication of how centralized a network is.
3. The average node degree is distributed similar to the actual graphs. Node degree is the number of links a node has. The average node degree we use is the average of the node degree of all nodes in a network.

For the majority of random graphs, we choose to generate them in such a way that they closely resemble values from the real world networks As noted above, these values are comparable to real networks found in the topology-zoo [7]. We have also generated some outliers, such as fully connected graphs and graphs that resemble a star topology to evaluate those specific scenarios.

4 Evaluation

To perform our evaluation we use the same approach and evaluation metrics for both network sets, and the three routing trees. In this section, we will present the methods and metrics we use, and how we use them to test the usefulness of the different approaches in a geocast scenario.

4.1 Evaluation Method

To evaluate multicast and geocast destinations in the graphs we use different methods. Both methods share the source node selection. Every node in the network is selected exactly once as the source for every possible destination set, the set of source nodes is equal to the set of nodes V. Runs are done for all destination sets containing 1 node, 2 nodes, up until the total number of nodes in the network (excluding the source). The destination set generation method differs between the multicast and geocast case.

Multicast: In the case of multicast these destinations are every possible combination of all other nodes in the network. If we take the example network given in Fig. 2, using node 0 as the source, the destination set would be {1, 2, 3, (1, 2), (1, 3), (2, 3), (1, 2, 3)}.

$$DS_{|V|-1}^s = \{\{d_1, ..., d_{|V|-1}\}, ...\} \in P(V - s)$$
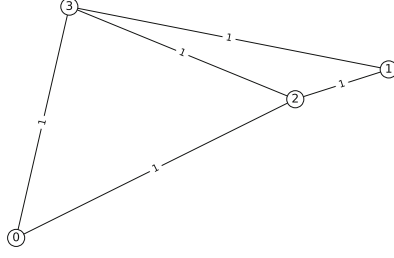
Fig. 2. 4 node example graph

The destination set $\mathbf{DS_n^s}$ is the set of all sets with length n from all distinct permutations of the node set without the source node $(\mathbf{V} - \mathsf{s})$. The maximum length of a destination set is $|V| - 1$, all nodes except the source node.

Geocast: The geocast evaluation selects each (non source) node as destination once and selects extra nodes that are geographically closest depending on the number of destinations required. For each of these destination nodes, 0 to N − 1 extra nodes are selected. The extra nodes are always selected based on their geographical distance, the first node added is always the closest, the second node is the second closest and so on. Destination sets are distinct, generated sets that are identical to already existing sets are ignored as they would represent the same geocast area. In the example network shown in Fig. 2 this would be {1, 2, 3, (1, 2), (3, 2), (1, 2, 3)} for source node 0. Note again that we do not use the same destination set twice here. In this case node 1 is also the closest other node to 2, we do not include (2, 1) as this will replicate (1, 2).

$$GS_{|V|-1}^{s,d} = \{d, v_1^d, v_2^d, ..., v_{|V|-1}^d\}|d, v \in (V - s)$$

$$GS_{|V-1|}^s = \bigcup_{d \in (V-s)} \{GS_{|V|-1}^{s,d}\}$$

In these equations $GS_{|V|-1}^{s,d}$ represents the geographic destination set with d as the initial destination and s the source, v_n^d are the other nodes in the network sorted by their geometric distance from d. $GS_{|V-1|}^s$ is the set of all distinct destination sets for source node s.

4.2 Evaluation Metrics

We evaluate the performance of the three different routing trees using the following metrics: (1) Path cost, (2) Edge usage.

To present the way we will interpret our graphs we will use the network in Fig. 1a as an example. This network has 11 nodes and 18 links.

The graphs used to present our results are generated using a consistent color coding scheme. *Blue* data belongs to the Shortest Path Tree, *red* data belongs to the Steiner heuristic and *green* data represents the Minimum Spanning Tree.

4.3 Path Cost

The average path cost in a network gives an indication of the cost to reach a number of destinations. We use all possible destination combinations to simulate multicast and clustered destinations for geocast.

$$GS_{|V|-1} = \bigcup_{s \in V} \{GS^s_{|V-1|}\}$$

$$DS_{|V|-1} = \bigcup_{s \in V} \{DS^s_{|V|-1}\}$$

We present the path cost as the average path cost for specific destination set sizes ($GS_{|V|-1}$ for geocast, $DS_{|V|-1}$ for multicast). This average is calculated over all possible source to destination trees with a certain destination size.

As an example with destination size 1: There are 11 nodes in the network shown in Fig. 1a. These 11 nodes each have 10 destinations giving us 110 (source, destination) sets. We take the average cost of these 110 routing trees for each of the three routing tree approaches.

We will start presenting our results as graphs that show the average cost for a number of destination per routing tree type. In Fig. 1b the results for network in Fig. 1a are shown. The error bars represent the standard deviation. For this specific network we can see that the Shortest Path Tree cost is close to that of the Steiner heuristic when the destination set is small. We can also observe that when the destination set includes all nodes the routing costs of all trees converge.

Later in the paper we show an average normalized path cost per graph. This cost has been normalized by the number of edges in a graph to allow comparison between graphs of different sizes.

4.4 Edge Usage and Fairness

To determine how 'fair' the link utilization is, we evaluate it for different networks. The link utilization metric describes how evenly the load is distributed in the network. If a few links are used for almost every combination of source and destination nodes it could get overloaded. Overloading a few links and leaving others completely unused is not likely to be a desirable property, and should be something to take into account.

We define edge usage as the normalized times per number of runs an edge was used when evaluating a graph. For example, if we did 10 runs and a certain edge was used in 6 of those runs, its edge usage would be 0.6. We believe the fairness of edge usage to be an important factor as it describes the load distribution within the network. A situation where few links carry almost all traffic might not be desirable from a cost and load distribution standpoint.

Using Fig. 3 we will explain how our stacked bar charts for edge usage are constructed. Figures 3a, b, and c show the edge usage fraction per edge for the network in Fig. 1a. Each of the bars represents an edge, with the height representing the fraction of runs this edge was included in the tree. These edges were

(a) Shortest Path   (b) MST   (c) Steiner Heuristic   (d) Combined
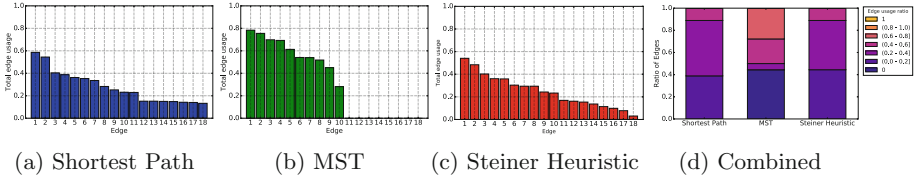
Fig. 3. Edge usage of the network in Fig. 1a

sorted with decreasing edge usage for viewing convenience. We can see that the Shortest Path and Steiner Heuristic trees use all edges and the Minimum Spanning Tree only uses 10 out of a total of 18. Figure 3d combines these graphs into a single stacked bar chart per routing tree. We can clearly see that the usage is more evenly distributed in the Shortest Path and Steiner Heuristic methods and that for the Minimum Spanning Tree a considerable fraction of edges is never used and another significant fraction is almost always in use.
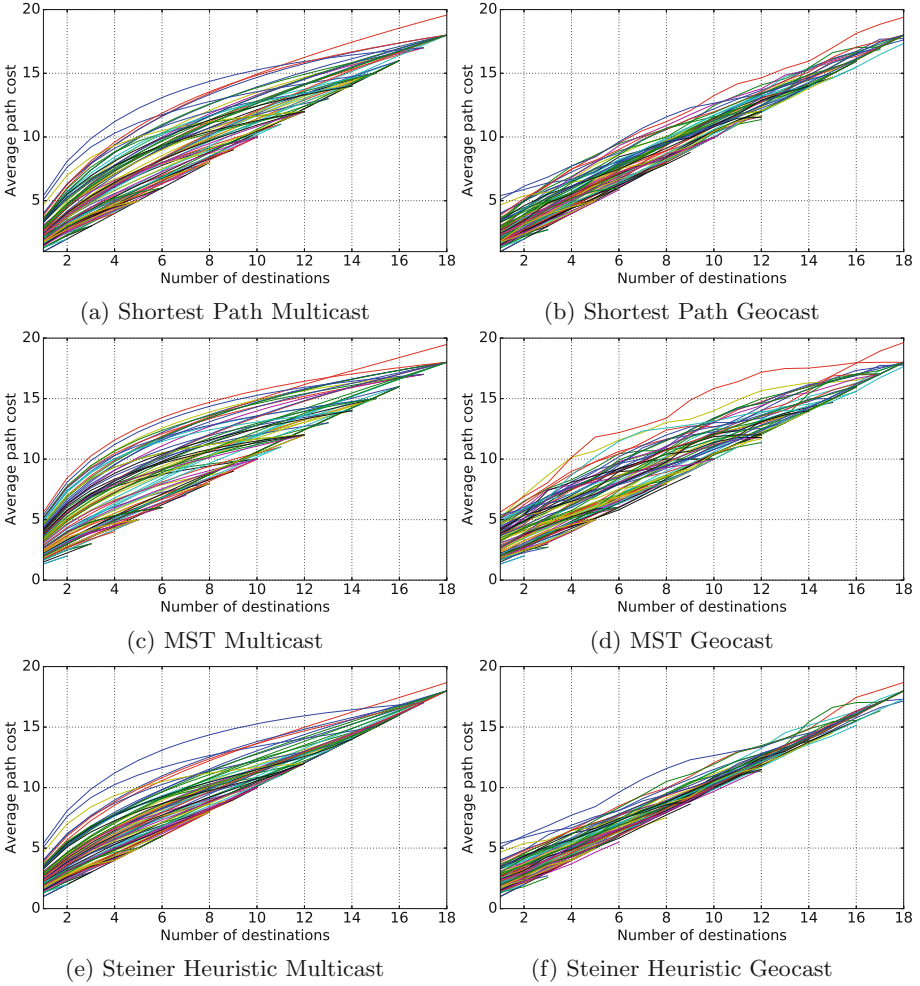
5 Results

In this section we will present the results over all the graphs we have evaluated. We start with the general results and go into specific cases later in the section.

5.1 General Results

Average Path Cost. As shown in Fig. 4, on average, almost all networks we evaluated show similar results. There are a few outliers visible in the results that we will discuss later. Figure 4 shows the results for the 85 networks taken from the Topology Zoo [7] that have less than 20 nodes with all edges having weight 1. It was not feasible to compute the multicast performance for the larger networks due to the large number of destination combinations. The graphs show the number of destinations on the x-axis (starting with 1) and the average cost on the y-axis. Each line in the graph represents one network.

In Figs. 4a and b we show the average cost of routing a packet in a multicast and geocast situation using Shortest Path forwarding on networks where all edges have cost 1. The case for using a Minimum Spanning Tree and a Steiner Heuristic can be seen in Figs. 4c, d and Figs. 4e, f respectively.
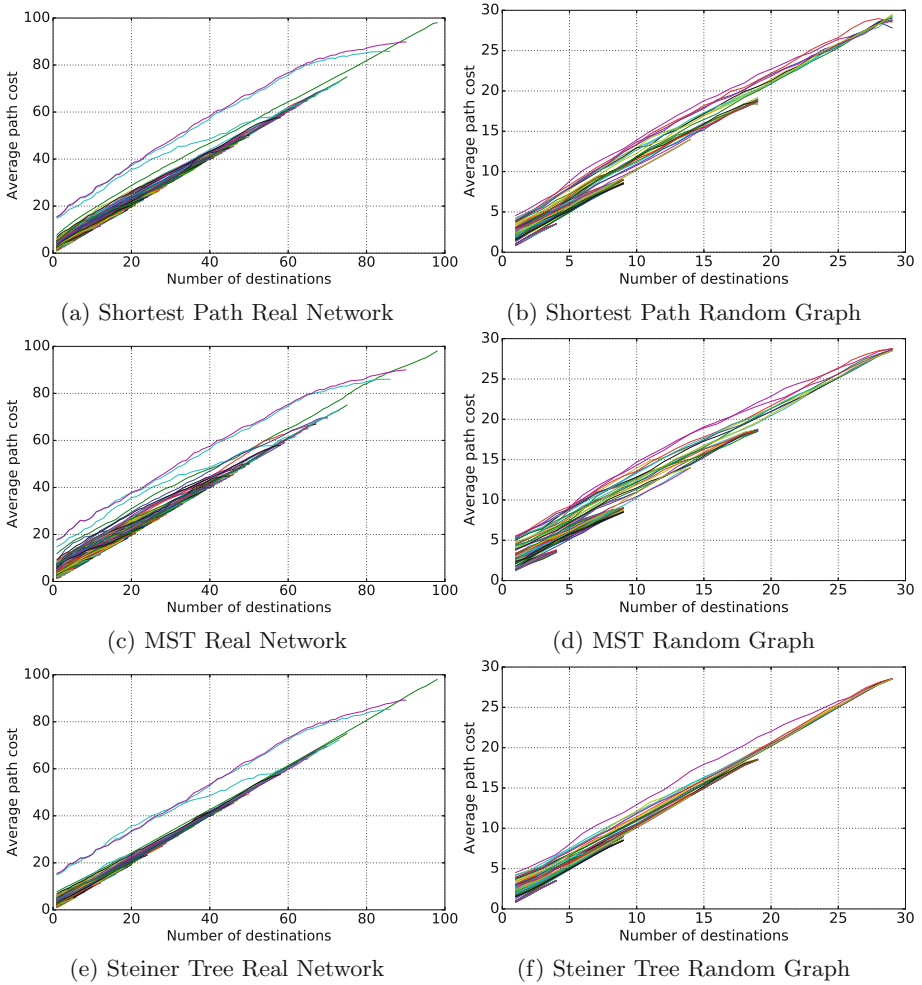
In general, we can see that the geocast scenario is more efficient in terms of forwarding cost than multicast in situations where the number of destinations is around a third of the total number of nodes in the network. We can also observe that the Steiner Heuristic is the most efficient forwarding method as expected. The Shortest Path method is however not that much less efficient while using significantly less computational resources. We can see that the Minimum Spanning Tree approach shows less than optimal results but is not necessarily much less efficient depending on the network. It also has the benefit of being precomputed so forwarding costs would be extremely low. The more well connected a

(a) Shortest Path Multicast

(b) Shortest Path Geocast

(c) MST Multicast

(d) MST Geocast

(e) Steiner Heuristic Multicast

(f) Steiner Heuristic Geocast

Fig. 4. Results over 85 real networks smaller than 20 nodes

graph is, the smaller the cost difference of geocast compared to general multicast becomes. In less well connected graphs that are more common in real world networks and in extreme cases such as line topologies, the geocast scenario is most optimal.

We evaluated geocast results for all networks in the Topology Zoo [7]. These graphs are an extension of the graphs in Fig. 4, also including the networks with more than 20 nodes found in the Topology Zoo. These results can be seen in Figs. 5a, c and e for the Shortest Path Tree, Minimum Spanning Tree and Steiner tree respectively. In these graphs we can see the linear relation between the number of destination nodes and the average cost more clearly. On average the number of nodes is equal to the average cost (or links used) to reach them for

(a) Shortest Path Real Network

(b) Shortest Path Random Graph

(c) MST Real Network

(d) MST Random Graph

(e) Steiner Tree Real Network

(f) Steiner Tree Random Graph

Fig. 5. Geocast results over 225 real networks and 98 random graphs

Shortest Path and close to optimal Steiner heuristic. The three obvious outliers here are networks that consist of several rings of large amounts of nodes. This leads to high overall cost to reach these destinations unless a significant portion of the network is used as destination.

Figures 5b, d and f show the geocast cost for the random geometric graphs we evaluated. These results are comparable to the results for the actual networks. We only observe a small difference in the lower maximum costs found, likely caused by the stronger correlation between geographic distance and network distance in the random geometric graphs.
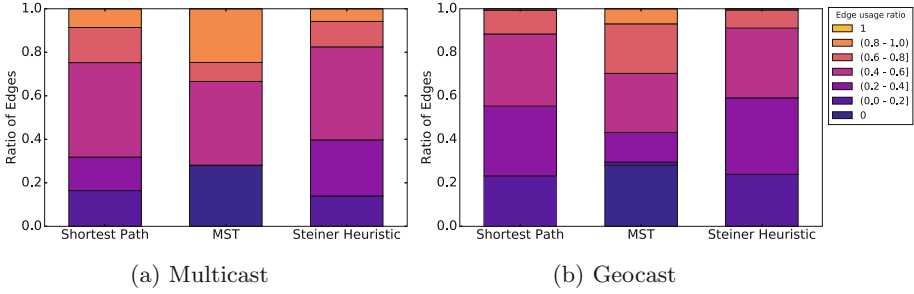
Fig. 6. Edge usage

Edge Usage and Fairness. In an ideal environment we would like to distribute the distribution tree in the network in such a way that every edge is used equally. This is under the assumption that (source, destination) pairs are also evenly distributed throughout the network.

In Fig. 6 we show the fraction of runs that a certain fraction of edges has been used. Each graph shows the results for Shortest Path, Minimum Spanning Tree, and Steiner Heuristic. In Fig. 6a and b we compare the results for all multicast runs with geocast runs over the same set of networks with less than 20 nodes. We can clearly see the difference between the multicast and geocast scenario. With multicast there is a number of edges that are almost always used, while this effect is diminished when destinations are geographically clustered. We observed the same results for geocast on the full set of real networks.

In general we can conclude that the fairness of the Minimum Spanning Tree approach is the lowest as a significant number of edges is never used. Of course this result was to be expected as the same tree is used for every (source, destination) set.

Generally, we observe that the more connected a network is the more equal the load is distributed. This makes sense as there are more possible paths in the network to reach all destination. On average multicast forwarding seems to use more edges compared to geocast. This result can be explained by the geographic clustering of the destinations, making the path from source to destinations share more edges.

5.2   Correlation with Network Characteristics

Some network characteristics have influence on the performance of forwarding trees. In other words, the way some networks are designed lead to a certain forwarding performance and give them specific values for these characteristics. The characteristics of particular interest are the average node degree of the network and the betweenness. We calculate the average normalized path cost per graph for the following results. The path costs are normalized by dividing them by the number of edges present in the graph.

Node degree is the number of edges a node has. In the case of a fully connected network this is equal to $N_{deg} = |G| - 1$. The minimum node degree is 1, as can be found in a node that is only connected to a single other node (for example in a star topology). The average node degree of a network is simply the average of all node degrees in that network.
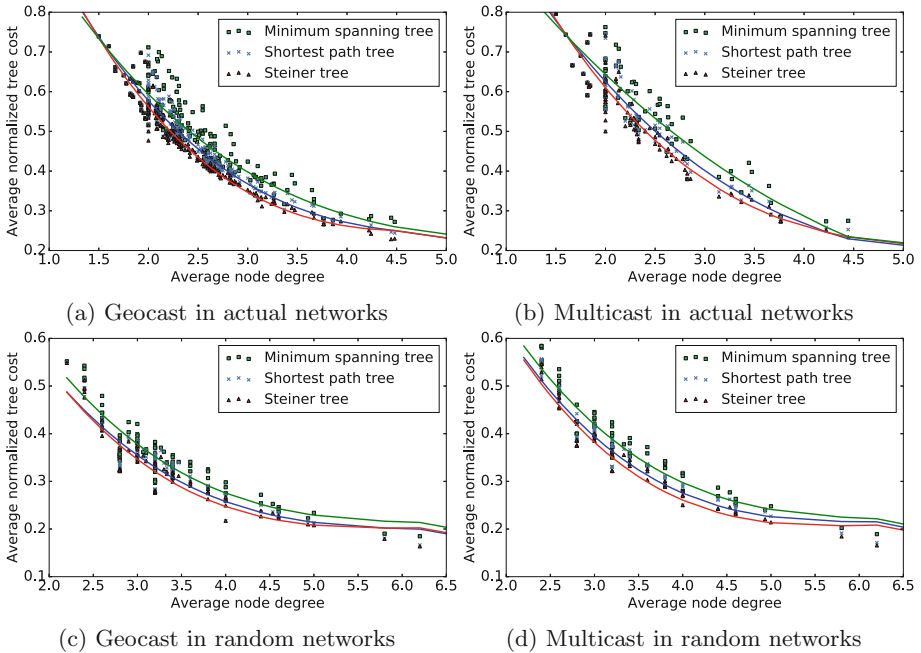
Figure 7 shows the normalized average path cost of a network for the different routing trees plotted against the average node degree of the network. As expected we see a strong correlation between the two values. We can conclude that the different routing trees converge when the node degree is higher. When the node degree is lower, more efficient forwarding trees are more beneficial to use.

The betweenness centrality of a node is the fraction of shortest paths the node is on in the network. Figure 8 shows that when the average betweenness centrality of a network is high, the average normalized path cost is also higher.
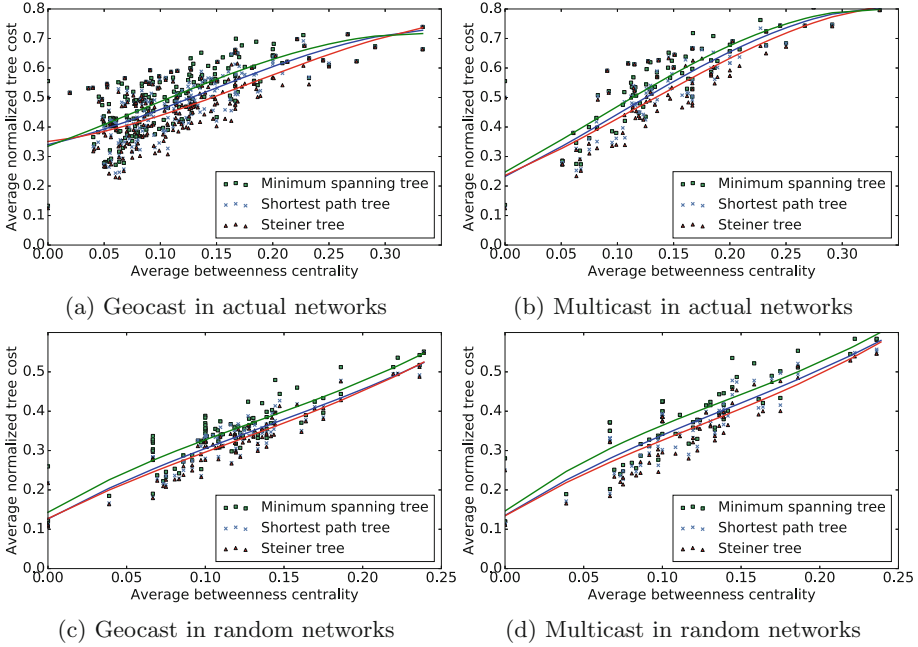
5.3   Special Networks

As mentioned before, the general topology of a network has a large effect on how efficient geocasting is in the network. A few types of networks that occur in the real world give interesting results. The shape of these networks might affect the choice of routing method that should be used in those networks.

We show the results of some these networks in Table 1. In this table we present the average link cost as fraction of the Steiner tree cost. We generated

(a) Geocast in actual networks   (b) Multicast in actual networks

(c) Geocast in random networks   (d) Multicast in random networks

Fig. 7. Node degree against cost

(a) Geocast in actual networks

(b) Multicast in actual networks

(c) Geocast in random networks

(d) Multicast in random networks

Fig. 8. Average betweenness against cost

networks off each network type 'Line', 'Ring', 'Star' and 'Fully connected' with 5, 10 and 20 nodes. In Table 1 these are shown as 'L', 'R', 'S' and 'F' followed by the number of nodes.

'Line' Networks: These networks simply look like strings with routers on them. Due to every router only having one link towards the geocast region in most cases the Shortest Path approach is very efficient here. In a 'true' line network the Minimum Spanning Tree is identical to the network and performs the same as Shortest Path and the Steiner heuristic as can be seen in Table 1.

Ring Networks: In networks that are designed as a ring the Shortest Path method is less efficient. This is likely caused by using both sides of the ring to reach a geocast area if the source is located on the opposite side of the destination in the ring. The Steiner heuristic always produces an optimal tree in such a network while the Minimum Spanning Tree can be extremely suboptimal depending on the destination nodes.

Star Networks: These networks generally have one or more hubs that have the majority of other routers connected to them in a star pattern. The effect is a few heavily used links between the hubs. If we consider a network that has only one hub we see that there is no difference in the performance between multicast and geocast routing. This makes sense as all routers (excluding the hub router) are two hops away from every other router (again excluding the hub). There is no

Table 1. Relative tree cost for special networks

Graph	L 5	L 10	L 20	R 5	R 10	R 20	S 5	S 10	S 20	F 5	F 10	F 20
SPT	1.0	1.0	1.0	1.081	1.155	1.191	1.0	1.0	1.0	1.0	1.0	1.0
ST	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
MST	1.0	1.0	1.0	1.205	1.253	1.291	1.0	1.0	1.0	1.215	1.170	1.132

possibility to optimize the distribution tree in this situation, every tree performs identical as seen in Table 1.

Fully connected Networks: An unlikely network to occur in reality, but an interesting theoretical situation to evaluate is the fully connected network. Here every router has a direct link to every other router. The result is a network in which every node can reach every other node in one hop. The Shortest Path and Steiner tree are always optimal (and identical) in this situation. The Minimum Spanning Tree will lead to two hops between most node pairs as it creates a star network. This result logically corresponds to the node degree graph, the higher the node degree (equal to $|V| - 1$ in this case) the lower the average cost.

6   Conclusion and Future Work

We set out the find the efficiency and fairness of Shortest Path, Steiner tree and Minimum Spanning Tree forwarding for geocast packets.

Based on our results we can conclude that for a relatively small number of destination nodes the Minimum Spanning Tree approach is the least efficient, using more edges and having, on average, a larger total cost. The differences between the Shortest Path Tree and Steiner tree is visible for small numbers of destinations but it is not that great.

We have shown that the average cost of a routing tree towards a geographically scoped destination is lower than that of a randomly distributed destination set. This result can be explained by the relation between geographical distance and network distance. The effect is most visible when the number of destinations is close to a third of the number of nodes in a given network.

We have also shown that a Steiner tree shows the most equal distribution of edge usage, closely followed by the Shortest Path Tree. As expected the Minimum Spanning Tree does not perform favorably on the edge usage metric due to the fixed distribution tree used. We do note that this behavior might be desired in certain situations.

It seems that networks with a high average node degree and low average betweenness centrality have the lowest forwarding costs. These characteristics can be used when deciding on a routing tree to use in a specific network.

Overall we conclude that a Shortest Path Tree is the most efficient choice for a geocast routing algorithm. Its performance and link fairness are close to that of the Steiner tree while requiring less computational resources.

For future work we will use the outcomes of this evaluation in the design of a routing algorithm for geocast based on the addressing scheme we developed earlier [5]. We will attempt to develop a Shortest Path geocast routing algorithm.

References

1. Karagiannis, G., Heijenk, G., Festag, A., Petrescu, A., Chaiken, A.: Internet-wide geo-networking problem statement (2013). https://tools.ietf.org/html/draft-karagiannis-problem-statement-geonetworking-01

2. Navas, J.C., Imielinski, T.: GeoCast - geographic addressing and routing. In: Pap, L., Sohraby, K., Johnson, D.B., Rose, C. (eds.) MOBICOM, pp. 66–76. ACM (1997)

3. Doar, M., Leslie, I.: How bad is naive multicast routing? In: Proceedings of the Twelfth Annual Joint Conference of the IEEE Computer and Communications Societies. Networking: Foundation for the Future (INFOCOM 1993), pp. 82–89. IEEE (1993)

4. Salama, H.F., Reeves, D.S., Viniotis, Y.: Evaluation of multicast routing algorithms for real-time communication on high-speed networks. IEEE J. Sel. Areas Commun. **15**(3), 332–345 (1997)

5. Meijerink, B., Baratchi, M., Heijenk, G.: An efficient geographical addressing scheme for the internet. In: Mamatas, L., Matta, I., Papadimitriou, P., Koucheryavy, Y. (eds.) WWIC 2016. LNCS, vol. 9674, pp. 78–90. Springer, Cham (2016). doi:10.1007/978-3-319-33936-8_7

6. Nguyen, U.T., Xu, J.: Multicast routing in wireless mesh networks: minimum cost trees or shortest path trees? IEEE Commun. Mag. **45**(11), 72–77 (2007)

7. Knight, S., Nguyen, H., Falkner, N., Bowden, R., Roughan, M.: The internet topology zoo. IEEE J. Sel. Areas Commun. **29**(9), 1765–1775 (2011)

8. Kou, L., Markowsky, G., Berman, L.: A fast algorithm for Steiner trees. Acta Informatica **15**(2), 141–145 (1981)

9. Hagberg, A.A., Schult, D.A., Swart, P.J.: Exploring network structure, dynamics, and function using NetworkX. In: Proceedings of the 7th Python in Science Conference (SciPy 2008), Pasadena, pp. 11–15, August 2008

10. Freeman, L.C.: A set of measures of centrality based on betweenness. Sociometry **40**, 35–41 (1977)