CGGC: Cached Greedy Geocast

Christian Maihöfer, Reinhold Eberhardt, and Elmar Schoch

DaimlerChrysler Research And Technology,
Communication Technology (RIC/TC)
P.O. Box 2360, 89013 Ulm, Germany
{christian.maihoefer|reinhold.eberhardt|elmar.schoch}@daimlerchrysler.com

Abstract. With cached greedy geocast, we propose an enhanced forwarding scheme for geocast especially in highly mobile ad hoc networks. We will show, that a cache for presently unforwardable messages during what we call the line forwarding, can significantly improve the geocast delivery success ratio, especially in sparse peopled networks with high node velocities. Therefore, we introduce and examine two separate caching strategies and present simulation results in conjunction with geocast. Our simulations inquire the delivery success ratio but also the trade–off in delivery delay.

1 Introduction

For all routing protocols in mobile ad-hoc networks, the major challenge is to find a route from the sender to the destination without any preconfigured information and under constantly varying link circumstances. In contrast to topology–based routing, the approach of position–based routing relies on geographic position information to deal with this problem. This means that all routing decisions, for example to which next node the packet should be forwarded, are based on the geographic destination data included in the packet. We distinguish between two basic classes of geographic forwarding: hop-to-multihop and hop-to-hop. In hop-to-multihop protocols one packet is sent to more than one next node, which means that the packet has to be duplicated and possibly reaches the destination more than once. This routing scheme results in a redundant delivery to a single destination or in the delivery to multiple destinations. In the other class the packet is routed only on exactly one, distinct way, resulting in a unicast delivery.

This paper considers geocasting, which is the transmission of a message to some or all nodes within a geographical area, called destination region. Geocast in position aware networks allows promising new services and applications. For example, in the automotive domain we aim at realizing virtual warning signs. Virtual warning signs are displayed inside the car, e.g. on the dashboard and can warn the driver about an accident after a blind corner, about an icy road, a wrong-way driver, a speed limit, and so forth. Virtual warning signs can easily be realized with geocast.

When designing protocols for the automotive domain, we must especially take care to be able to cope with high node velocities, which is typically not considered in related work about ad hoc networks. Our proposed cache scheme is an extension of regular geocast protocols in order to deal with this situation of high velocities, i.e. constant neighbourhood changes and unstable routing paths.

The geocast protocol that we rely on and extend in this paper uses both routing schemes explained above. As long as a packet is outside the destination region of a geocast packet, a network efficient hop-to-hop delivery is applied. Inside the geocast destination region, hop-to-multihop routing allows the delivery to all nodes of the destination region. In our geocast protocol [1, **?**] we divide the overall delivery process in exactly these two substantial parts, which on the one hand is called line–forwarding as long as the initial packet hasn't reached its destination region and on the other hand area–forwarding inside of this region.

To be able to select a single next node during line–forwarding, we use a beaconing system that allows constant neighborhood awareness. If a packet arrives, the routing layer on each relaying node is able to decide, based on the neighbored nodes' and the packets' destination position information, to which particular neighbor a message should be forwarded.

The most obvious decision scheme in this context is greedy forwarding. When using it, a message is always forwarded to that neighbor, whose position is nearest to the packet's destination, which is primarily defined through a position vector. This procedure should guarantee that a message is routed to the destination position in a minimum number of hops. But greedy forwarding has also one considerable disadvantage. If a node tries to forward a message while no neighbor is present that is closer to the destination than the relaying node himself — we call it a local maximum or dead end — this node is not able to continue the greedy forwarding at this point in time. In more detail, a *local maximum* is reached at a node n if no neighboring node of n is closer to the destination region than n. In case a local maximum is reached, the routing protocol must provide a recovery strategy if the message shall not be lost.

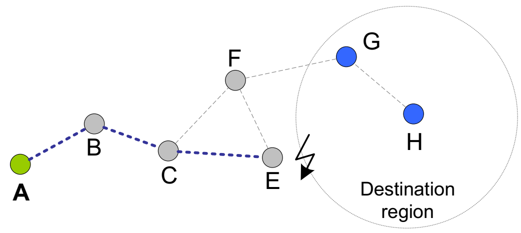

Fig. 1. Local maximum prevents successful forwarding

In Figure 1, node A sends a message to the depicted destination region using greedy forwarding. The described problem arises when the message reaches node E. Although there would be a logical path into the destination region through nodes A-B-C-F-G, simple greedy forwarding fails to deliver the message. For geocast it is especially reasonable to try to achieve a maximum of successful line–forwardings, since each loss of a packet that doesn't reach the destination region prevents a successful delivery to any node inside of it.

In order to overcome these flaws, we propose cached greedy geocast. The main idea of cached greedy geogast is to add a small cache to the routing layer that holds those packets a node cannot forward instantly due to a local maximum. As we use a beaconing subsystem anyway to be able to discover and hold a table of neighbored nodes, this cache is notified whenever a new neighbor comes into reach or an already known neighbor changes its position. Then the cache can check all currently stored messages whether they can be forwarded to a newly discovered or moved neighbor, because those neighbor's position is now possibly closer to a geocast destination region than the current nodes position.

Please note, our proposed caching scheme is especially designed for the use in ad hoc networks with high velocities and having certain applications in mind like the mentioned virtual warning sign. In particular, it is not meant to use that scheme for regular unicast delivery, since higher layer protocols like TCP are likely to result in degraded performance when caching is applied.

The overall paper is organized as follows: Section 2 discusses other approaches in the context of position–based routing, especially concerning geocast. In Section 3, we describe our proposed ideas and our implementation of cached greedy geocast. Afterwards, we will show simulation results of our algorithms and conclude the work with a brief summary.

2 Discussion of related work

For the classic definition of geocast, meaning that information is delivered to all nodes currently residing within a specific geographical region, quite a number of approaches have been proposed. Where the distribution of data inside the target area is mostly done in the same way by applying a flooding mechanism, the methods to transport data packets to that region differ elementarily. Basically, there are three methods implemented: restricted or modified flooding, hierarchical and greedy forwarding. For a complete overview of geocast protocols see [**?**].

Flooding derived protocols like Location Based Multicast (LBM) [2] or GeoGrid [3] usually define a virtual geographic region, inside of which every contained node forwards a packet, if it hasn't done so yet. Obviously, the protocol needs a sequence number mechanism to be able to identify packets that already have been received and forwarded at a node before. Because the forwarding region also includes the final destination region, it is assured that the data packets reach all concerned nodes within this destination region. A node residing inside the destination region then can pass the data to higher layers. The advantage of that delivery process is redundancy, since messages are able to reach the destination on many ways in parallel. This, of course, results in high network load typicyal for flooding mechanisms.

A completely different approach is to apply a hierarchical forwarding. In [4], an approach is proposed which uses geographic information for routing in a fixed network like the internet. The GeoNode protocol assumes that the network has a cellular structure. For each cell, there is a node called GeoNode that relays all messages for the nodes inside the cell. Position–aware GeoRouters take care to distribute data to other GeoRouters according to the packets' geographic destination information. If a node wants to send a message, it forwards it to the local GeoNode, which in turn passes it to the next GeoRouter. The GeoRouter then determines to which other GeoRouters the packet has to be forwarded in order to cover the complete destination region.

The other approach in this context is GeoTORA [5]. The basic idea of this approach is the usage of an acyclic, directed graph that allows to route the packet by corresponding graph algorithms. GeoTORA maintains such a graph for every intended geocast destination region. The graph representation is implemented by assigning a height value to every node on the network, where nodes inside the destination region carry a value of zero. All other nodes' height values are assigned according to the number of hops that are necessary to reach the destination region. When forwarding a message, a node always transmits it to a neighbor with smaller height.

As we already described, we used the third possibility, greedy forwarding, for our own implementation. Reasons for this decision are that it neither has the effect of the high network load caused by a flooding approach nor the necessity of building up a hierarchical structure. This makes it a good candidate for our considered highly mobile ad hoc networks.

Nevertheless, pure greedy forwarding also has the flaw that it cannot reach a destination if a local maximum at a forwarding node is reached. This issue is targeted for example in GPSR [6]. GPSR is a position–based unicast routing protocol which generally applies greedy forwarding and introduces a so called "perimeter mode" to route around a gap in the network. As long as a packet cannot be forwarded to a node closer to the destination, it surrounds the network gap on a tangential way. The idea of perimeter mode is to forward the message according to the right-hand-rule. This means, node x selects those neighbor n as next hop, which is the first in counter-clockwise order to a thought line from node x to the destination position. This method is only used if no closer neighbor is available.

Our approach described in this paper also targets this greedy forwarding flaw by caching packets that cannot be forwarded instantly. This solution is especially effective in highly mobile and sparse peopled ad hoc networks, where the perimeter mode of GPSR is less effective (see simulation results in Section 4). As examined in [7], node mobility can increase the overall capacity of an ad hoc network. We will show that cached greedy forwarding can also utilize the mobility to significantly improve delivery success probability.

3 Overview of Cached Greedy Geocast

3.1 Detailed problem description

Geocast in its most general definition means to transport a packet from the sender to all nodes that currently are inside a given geographic region. We assume that the sender specifies the destination region he wants to send the packet to. Therefore, we only need position and neighborhood information for routing. The region definition is variable, but contains at least one point inside of the region, which we use as target for our routing. Usually, a circular area with its center and its radius is applied. As introduced before, we decided to use a greedy mechanism that routes a packet hop per hop in direction to the destination region, until the packet is received by a node that is inside this region. Then, an area–forwarding method, flooding, is applied to spread the packet among all nodes inside the region. The main focus of the paper is the line–forwarding part of geocast.

At first, simple greedy forwarding has some major disadvantages. Because the routing scheme relies on the availability of suitable neighbors, i.e. at least one neighbor that is closer to the destination, it is necessary to define what takes place if this elementary condition is not given. Usually, a packet is dropped under these circumstances. But this results in many losses, since network gaps, partitions, and outdated neighbor information are quite common in highly mobile scenarios. Moreover, although our geocast is a "best effort" service, we cannot accept such high loss probabilities, because any packet that is dropped during the line forwarding cannot reach *any* of its target nodes inside the destination region. On the other hand, using perimeter mode of GPSR, our simulation results (see Section 4) show that it is not effective with high velocities.

We define in the following two strategies which improve line–forwarding with the simple greedy scheme: enhanced and cached greedy forwarding. Both techniques profit explicitly from mobility, which we commonly face in our applications. The main focus of this paper is the caching approach, but we also describe and evaluate the enhanced forwarding shortly to be able to estimate its effects.

3.2 Enhanced greedy forwarding

A simple recovery strategy for greedy forwarding failures is to delay a packet a short period of time, and then retry to forward it. If this neighbor is no longer within the wireless transmission range, another neighbor is selected. This procedure may be repeated some times. If a suitable neighbor is in reach at any retry time, the packet is finally forwarded, or otherwise dropped, too. Although this is a quite simple scheme, it already helps in highly mobile ad hoc networks to improve delivery success ratio.

3.3 Cached geocast approach

In order to significantly achieve further reduction in experienced packet losses during line–forwarding, we propose a caching mechanism. The idea of cached geocast is to add a storage to the routing layer that takes up those packets, where the basic greedy schemes have failed. This cache of currently unroutable packets, called *LocalMaxCache* doesn't make any forward attempts himself, but gets notified about each newly discovered neighbor or about changes in neighbors' positions. Neighbors are discovered and maintained by a periodic beaconing system (see Figure 2), and the information in every beacon also contains the current position of the node the beacon comes from. Thereby, whenever having detected a new neigbor or changed relative positions to neighbors, the beaconing subsystem can notify the LocalMaxCache and provide the relevant position information. The cache checks, whether there is a packet stored, whose destination fits to the new node in that manner that the new node is closer to it (see Figure 3). This means, our cache operates on demand and does not result in higher network message overhead in contrast to blind periodical resending attempts. The complete forward algorithm in pseudocode is given in Figure 4.

We expect from the caching strategy to significantly increase geocast delivery ratio especially in highly mobile environments. A trade–off will be to experience higher end–to–end delays. However, note that only packets are delayed which are currently not forwardable and which would be dropped otherwise. This means, the cache does not increase delay for those packets that would be delivered with regular greedy routing, too.

3.4 Implementation description

We propose and investigate two separate alternatives for the line–forwarding cache. The first one consists of a size–restricted queue for packets, where *size*

```
N gets beacon (S, P):
// S...address of beacon sender, P...position of sender
// T...neighbor table
s := T[i].address == S;
if s == true then // neighbor already registered
    T[i].position := P; // update position
else //neighbor is new
    add (S,P) to T;
checkCache; // check cache for forwardable messages
```

Fig. 2. Pseudocode for beaconing system

```
N procedure checkCache:
// Q...LocalMaxCache queue, g...geocast message,
// (g.c...geocast destination region center)
// T...neighbor table
geocast message g, neighbor nb := ∀ i ∀ j: ||pos(T[j]), Q[i].c|| ≤ ||pos(N), Q[i].c||;
    send geocast g to p;
    remove g from Q;
```

Fig. 3. Pseudocode for checking the cache about forwardable packets

limits the number of packets stored and not the actual payload size. The length of the storage period is unlimited. The other approach hasn't a limitation of size, but packets will only be stored for a limited period of time.

Size restricted cache

After a packet has passed all attempts to forward it to a node that is closer to its destination region than the current relaying node, it is handed over to the cache. If the cache queue is already full, i.e. the maximum size is reached, the oldest packet in it is dropped which means that its second chance to reach the destination region ends at that time. It is obvious, that dropping packets at the end of the queue implicitly limits the caching time depending on how much traffic occurs. On the other hand, in low traffic situations, packets may remain

```
N forward packet (P):
// P.d...packet's destination region, P.d.c... destination region center
// P.fwdRetries...packet's forward attempts at current node
// maxFwdRetries...maximum forward retries
// T...neighbor table, nh...next hop
nh := findNearestNeighbor(P.d.c); // find neighbor closest to destination
if nh != NULL then // suitable neighbor was found
    send packet P to nh
else // no greedy neighbor available
    if P.fwdRetries < maxFwdRetries then
        delay packet P and recall forward(P);
    else
        put packet P into cache;
```

Fig. 4. Main forwarding mechanism

in the cache very long or even until the node is powered off. As long as a packet is stored in the cache, it is checked against every discovered neighbor, whether this one is a suitable next relay for it.

Time restricted cache
The time restricted cache works similarly to the size restricted cache but takes queuing time into account. A new packet is always added to the queue, no other packet has to be dropped. All added packets are stored for a certain period of time and dropped after the expiration of this interval. Thereby, we can specify a border beyond of which it doesn't make sense to forward a packet any more. Another aspect of this restriction alternative is the possibly unlimited growth of the cache. However, if we define a maximum caching time and given a known maximum sending and receiving rate, the maximum cache size is deterministic and known. Besides of that, packet forwarding out of the cache is done in the same manner as described before.

4 Simulation results

In order to evaluate our approach, we implemented our cached geocast in ns-2, and examined several preliminary expectations we had on the design as well as the behaviour of cached greedy geocast in contrast to GPSR, which implements a different meachnism to overcome local maximums.

4.1 Environment and setup parameters

For our simulations, we considered several parameters as important. The most significant of them are of course network size and node count, which define the effective node density. Commonly, quadratic network sizes between 1000 and 4000 meters side length, with a 500 m interval, and a node count of 100 are used. As a standard wireless transmission range 250 m were applied. Therefore, node density varies from an average of about 20 nodes in the wireless transmission range in a 1000x1000m network to statistically 1.2 nodes in the wireless transmission range when using a 4000x4000m network.

All nodes are initially distributed uniformly over the network and move according to the random waypoint model. As we deal with highly mobile networks, a maximum speed of 50 meters per second and a pause time of 0 are common to all simulations. Note that these network and mobility specifications represent quite hard conditions for an ad hoc routing protocol. But these conditions were chosen intentionally, since it is a goal of our approach to overcome and even profit from them. To proove this, we will also show a comparison of our approach and GPSR both with no node mobility, i.e. a static network.

Another relevant item is the traffic that is to be transmitted on the network. In every scenario, we send 100 messages. Both the sender and the destination region are selected randomly, whereby the destination region radius ranges from a minimum of 100 to a maximum of 300 meters. For the comparison simulations with GPSR, which needs a node address as destination, we randomly choose one node out of the destination region. Our implementation of geocast does not need a node address but only the destination region. As we use an average value of 20 single simulation runs with identical parameters except the traffic, all other effects to the results should be eliminated.

transmission range	$250m$
mobility model	random waypoint
maximum speed	$50m/s$
simulation time	$60s$
network size	$1000m$ to $4000m$ squares
node count	100

Fig. 5. Overview on basic simulation parameters

4.2 Evaluation results

The most interesting values to examine are the mainly targeted delivery ratio and the end–to–end delay.

Line–Forwarding examination
We start with simulations of pure line–forwarding without geocast delivery, i.e. unicast delivery, with different count of allowed forwarding retries if a suitable neighbor is not available. This addresses the enhanced forwarding and primarily doesn't touch the proposed caching. In this context two parameters are important: first one is the allowed retry count, second one is the interval between these forwarding attempts. To be able to estimate the resulting differences clearly, we chose one strict pattern similar to simple greedy forwarding that allows no retries. Besides that, we applied a scheme which allows 1 retry after 0.1 seconds of waiting, and one more lax pattern allowing 5 retries and 0.5 seconds between them.

As depicted in Figure 6, a resulting gain in delivery success ratio through more retries is shown.

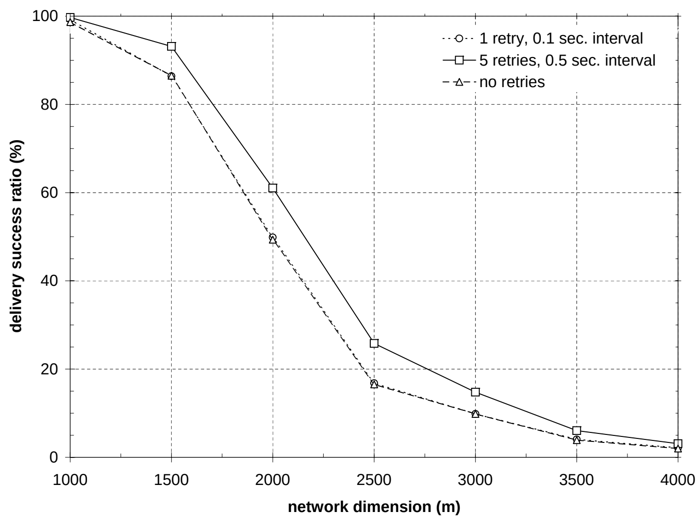

Fig. 6. Line–forwarding success rate results with different retry count and interval parameters

Having observed the effects of the enhanced greedy forwarding, we now focus on the cached forwarding mechanism. Therefore, we compare the line–forwarding as seen before with the cache scheme. A simulation result of GPSR, which has the same task as our line–forwarding, is given as a reference for comparison with our schemes.

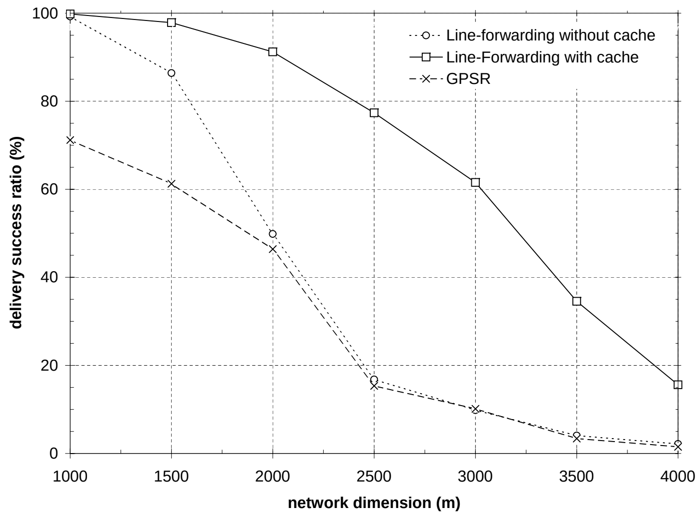

Fig. 7. Delivery success ratio comparison between standard line–forwarding, cached line–forwarding, and GPSR

In Figure 7, we see our expectations confirmed. The caching results in a significantly increased delivery success ratio in comparison to the ordinary line–forwarding. GPSR is already outperformed by our simple enhanced one retry line–forwarding in the more dense networks up to 2500 meters. The added caching outstrips both other schemes. Especially at network sizes over 2500 x 2500 meters, which are important for our applications as we assume that penetration rates may be low when introducing vehicular ad hoc networks, the caching approach performs very well.

For further investigation of the advantages and disadvantages of perimeter mode in constrast to our caching approach, it is necessary to examine the results of the opponent algorithms in mobile and non-mobile network scenarios. In Figure 8, it becomes clear that cached line–forwarding performs worse than GPSR with perimeter mode in non-mobile networks, since the cache works properly only with mobility. On the other hand, with the high velocities we assume for vehicular scenarios, cached line–forwarding reaches much higher delivery ratios than GPSR.

Geocast with and without cache
Until now we have only seen results concerning the line–forwarding, which is of course the phase of geocast, where the cache can unfold its function. As we propose the approach in the context of geocast, we will now take a closer look on the impacts of caching on geocast.

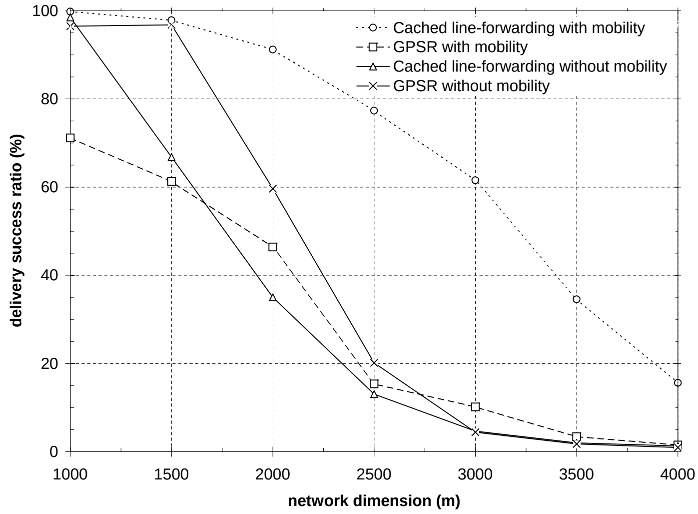

Fig. 8. Delivery success ratio of GPSR and cached line–forwarding in highly mobile as well as immobile ad hoc networks

We compare a standard geocast using only the enhanced forwarding scheme with our two caching approaches, size–restricted and time–restricted cache. In Figure 9, we display the simulation results.

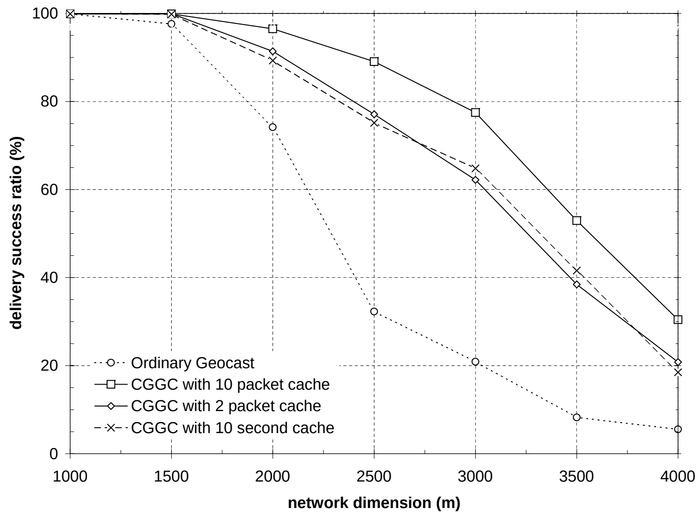

Fig. 9. The number of successful geocasts is increased by caching during the line–forwarding. The cache with 10 packets capacity performs best.

These results allow several conclusions: Geocast profits from the gain in successful line–forwardings in the same magnitudes as line–forwarding alone. We

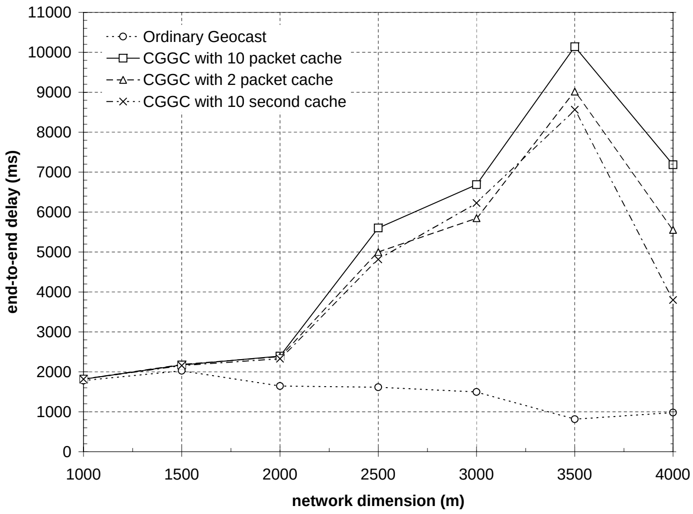

Fig. 10. Delays of cached geocast alternatives vs. ordinary geocast show the effect of caching.

see that the time–restricted cache performs slightly worse in this scenario than the size–restricted one. In these simulations, we used 10 packets as maximum size–restriction and a caching time of maximum 10 seconds on the other variant, respectively. Another very interesting aspect is the significant improvement of a small cache with only 2 packets compared with standard geocast. This means, even such a small cache is quite effective.

Concerning the protocol overhead, which we measured by the sent messages, there are only minimal effects of the caching scheme, since the cache is a on-demand mechanism and simply utilizes the beaconing that is running anyway.

Figure 10 shows another expected result of the cached greedy geocast approach: end–to–end delays are significantly increased. There is clearly a trade–off between delivery ratio and delay results — the implementation with the highest delivery ratio also shows the highest delay values, which of course is caused by the time a packet rests inside the cache. But it is important to realize that these high delays are average values caused by those packets that reach their destination region finally only due to the caching. From this point of view, the caching doesn't result in higher delays for those packets that do not need it, but increases the overall delivery ratio. Please note that smaller delay results for the 4000 m square network are caused by the lack of a representative number of successful deliveries.

5 Summary and outlook

We have presented cached greedy geocast, a mechanism to improve greedy forwarding that is especially suitable for ad hoc networks with high mobility and high node velocities and low node densities, i.e. sparse networks. The work shows that both alternatives, size restricted and time restricted caching, can significantly increase the messages successfully delivered to a geographic destination region.

As a trade–off, we experience higher average end–to–end delays with caching. This is because more messages reach their destination region. However, these additional delays are only caused by those packets that reach their destination region only due to the caching mechanism and which would be dropped, otherwise.

As network load is concerned, the caching nearly doesn't have any effects, since it is a on-demand mechanism. The load is increased only slightly, because more line–forwardings can be continued successfully.

References

1. C. Maihöfer, Walter Franz, and Reinhold Eberhardt, "Stored geocast," in *Proceedings of Kommunikation in Verteilten Systemen (KiVS)*, Leipzig, Germany, Feb. 2003, pp. 257–268, Springer Verlag.
2. Young-Bae Ko and Nitin H. Vaidya, "Geocasting in mobile ad hoc networks: Location-based multicast algorithms," in *Proceedings of the 2nd Workshop on Mobile Computing Systems and Applications (WMCSA 99)*, New Orleans, USA, Feb. 1999, pp. 101–110.
3. W.-H. Liao, Y.-C. Tseng, K.-L. Lo, and J.-P. Sheu, "GeoGRID: A geocasting protocol for mobile ad hoc networks based on GRID," *Journal of Internet Technology*, vol. 1, no. 2, pp. 23–32, Dec. 2000.
4. T. Imielinski and J. Navas, "GPS-based addressing and routing," Internet Engineering Task Force, Network Working Group, Request for Comments, RFC 2009, Nov. 1996.
5. Y.-B. Ko and N. H.Vaidya, "GeoTORA: A protocol for geocasting in mobile ad hoc networks," in *Proceedings of the 8th International Conference on Network Protocols (ICNP)*, Osaka, Japan, Nov. 2000, pp. 240–250.
6. B. Karp and H.T. Kung, "Greedy perimeter stateless routing for wireless networks," in *Proceedings of the Sixth ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000)*, Boston, USA, Aug. 2000, pp. 243–254.
7. M. Grossglauser and D. Tse, "Mobility increases the capacity of ad-hoc wireless networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, Aug. 2002.