Beaconless geocast protocols are interesting, even in 1D

Joachim Gudmundsson* Irina Kostitsyna† Maarten Löffler‡ Vera Sacristán§ Rodrigo I. Silveira§

Abstract

Beaconless geocast protocols are routing protocols used to send messages in mobile ad-hoc wireless networks, in which the only information available to each node is its own location. Packets get routed in a distributed manner: each node uses local decision rules based on the packet source and destination, and its own location. In this paper we analyze some of the most relevant existing protocols in a formal and structured way, focusing on two relevant 1D scenarios.

1 Introduction

In mobile ad-hoc wireless networks there is no fixed infrastructure or global knowledge about the network topology. Nodes communicate on a peer-to-peer basis, using only local information. Thus messages between nodes that are not within range of each other must be sent through other nodes acting as relay stations. An important particular case of ad-hoc wireless networks are wireless sensor networks, in which a (usually large) number of autonomous sensor nodes collaborate to collectively gather information about a certain area.

Nodes are typically mobile devices whose location and availability may change frequently, resulting in a highly dynamic environment in which routing must be done on-the-fly. Typically, messages are not sent to a particular network address, but to some or all nodes within a geographic region. This is called *geocasting*. The main pieces of information used to send a message are the location of the source node, and that of the destination region (also referred as *geocast region*), which is usually included in the actual message.

Many geocast protocols have been proposed. In general, existing protocols can be divided into two groups: those that assume that each node also knows the location of its 1-hop neighbors (i.e., all nodes within range) and those that don't. In practice, the locations of neighbors can be obtained by regularly exchanging *beacon* messages in the neighborhood. Beacons imply a significant message overhead, which prevents these methods from scaling even to medium-size networks [2]. For this reason, in this paper we are interested in the second group, the so-called *beaconless* geocast protocols.

Probably the most straightforward beaconless geocast protocol is simple flooding: each message is broadcasted to all neighbors, who in turn broadcast it to all their neighbors, and so on. Even though it is effective, the resulting message overhead is clearly unaffordable. From there on, there have been many improvements proposed. The ultimate goal is to reduce the message overhead as much as possible while still guaranteeing delivery. Due to space limitations, a proper review of all existing beaconless geocast protocols is not possible here. We refer the reader to [4] for a survey. Given the importance of geocast protocols and the many options available, several comparative studies have been presented (e.g., see those in [1, 3]) to asses the efficiency and efficacy of different methods under different scenarios. However, previous comparisons are mostly based on computer simulations.

In this paper we are interested in analyzing the behavior of beaconless geocast protocols from a theoretical and geometric perspective, since the geocast problem is inherently geometric. To that end, we present a structured overview of the main existing protocols in the literature, and identify important quality criteria to analyze them mathematically. The behavior of a geocast protocol, in general, must be analyzed in the context of a particular geometric scenario (i.e., a certain configuration of nodes). In this paper, we focus on two very fundamental geometric scenarios in 1D. Even though it is clear that the full complexity of these protocols can only be appreciated in two dimensions, we show that the 1D scenarios considered, despite their apparent simplicity, already pose interesting challenges, and already expose many of the essential differences between the protocols studied. For each scenario we analyze worst- and expected-case performance of six different protocols. The results obtained corroborate many of the findings previously obtained by simulations, and gives some insight into the difficulties of the 2-dimensional case.

2 Studied protocols

We have analyzed a selection of beaconless geocasting heuristics that are representative of different widely used strategies. These heuristics are frequently com-

*School of Information Technologies, University of Sydney, joachim.gudmundsson@gmail.com

†Dept. of Mathematics and Computer Science, Eindhoven University of Technology, i.kostitsyna@tue.nl

‡Dept. of Information and Computing Science, Universiteit Utrecht, m.loffler@uu.nl

§Dept. de Matemàtiques, Universitat Politècnica de Catalunya, {vera.sacristan,rodrigo.silveira}@upc.edu

This is an extended abstract of a presentation given at EuroCG 2016. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

bined in geocasting protocols, where the nodes follow local decision rules that are disjunctions of several different heuristic predicates. In addition, the resulting predicate is often combined via conjunction with a location predicate to control the region where each packet must travel. In this section we sketch the main characteristics of each geocast heuristics.

Simple flooding. In this protocol, when a node receives a packet, it broadcasts it—after checking that it has not broadcasted it before—and stores its ID in order to make sure it will not broadcast it again. This strategy is simple and robust, but it is non-scalable, as it produces an excessive and unnecessary network load. In the following, we describe several heuristics intended to reduce such flood load. Nevertheless, it is interesting to consider simple flooding not only for comparison purposes, but also because it is used as a building block in other protocols [3].

2.1 Restricted flooding

The following are two simple heuristics that can be considered restricted versions of flooding.

M heuristic. The MinTrans (M) heuristic explicitly controls redundancy through a parameter M: A node broadcasts a received packet if and only if the number of transmissions received for that ID is less than M. The redundant propagation allowed by the parameter M helps against problems such as message collisions and getting out from local optima.

T heuristic. The Threshold (T) heuristic uses location information for spreading the geocast propagation outward: A node retransmits a received packet if and only if the closest among all transmitters of packets with the same ID is at least a distance T from it.

2.2 Distance-based heuristics.

The previous heuristics are likely to have delivery failures in the presence of obstacles. The following protocols were designed to help solving this problem.

CD heuristic. The Center-Distance (CD) heuristic [1] relies on proximity: A node retransmits a received packet if and only if its distance to the center of the geocast region is less than that of all originators of transmissions received for the packet ID.

CD-P heuristic. This protocol [1] uses priority queues in order to further reduce the scalability problems of the CD heuristic. Each time the node can transmit, it transmits any packet that has not been transmitted at all yet (if any) or it (re)transmits, among all heard packets, the one whose transmission would give the largest reduction in distance to the center of the geocast region.

2.3 Delay-based heuristics

Some strategies to further reduce redundancy combine distances with retransmission delay.

BLR heuristic. In the Beacon-Less Routing (BLR) heuristic, each node determines when to retransmit a received packet based on a dynamic forwarding delay function valued in $[0, \mathrm{MD}]$, for MD a constant representing the maximum delay. The node retransmits the package after such delay, unless some other node does it before, in which case the retransmission is canceled. Three delay functions have been suggested in [2], based on the following parameters: r (transmission range), p (progress towards destination of the orthogonal projection of the current node onto the previous_node-destination line), and d (distance from current node to the source-destination line). The proposed variants are: $\mathrm{delay}_1 = \mathrm{MD}\,\frac{r-p}{r}$, $\mathrm{delay}_2 = \mathrm{MD}\,\frac{p}{r}$, and $\mathrm{delay}_3 = \mathrm{MD}\,\frac{e^{\sqrt{p^2+d^2}}}{e}$.

GeRaF heuristic. Based on distance, the Geometric Random Forwarding (GeRaF) protocol [6] logically divides the area around the destination of a packet p into n_p areas $\mathcal{A}_1, \ldots, \mathcal{A}_{n_p}$, where in $\mathcal{A}_1$ are all nodes closest to the destination, and so on. Once p is transmitted, up to n_p phases start, during which all nodes listen during a fixed amount of time. In the first phase, nodes in region $\mathcal{A}_1$ get to reply. If only one node replies, then that one will forward the message. If there are more, some collision resolution scheme must be used. If there is no reply, then it is the turn to reply for nodes in region $\mathcal{A}_2$. This process continues until some node in the non-empty region closest to destination replies.

Greedy routing (beaconless version). Greedy routing does not always guarantee delivery. Nevertheless, a greedy routing strategy is often used as building block of geocast protocols. For this reason we also consider greedy routing in our analysis. One example is Geographic Distance Routing (GeDiR) [5]. GeDir requires to know the position of all neighbors of a node: it is a greedy algorithm that always forwards the message to the neighbor of the current node whose distance to the destination is minimum.

This strategy can be made beaconless by a delay function based on the following parameters: r (transmission range), d (distance from previous node to destinations), and x (distance from current node to destination), by using delay function $\mathrm{delay}_4 = \mathrm{MD}\,\frac{x+r-d}{2r}$.

This strategy tries to get out of local minima by sending the packet to the best positioned neighbor, even if it is not closer to destination than the sender.

All protocols described in this section include a rule that states what to do if a node receives a message already in its queue. One option, like in BLR, is to always cancel the transmission of a message received

Figure 1: Illustration of the scenarios. Left: with unbounded reach, the k messages arrive immediately to all nodes, but that does not prevent intermediate nodes from forwarding the messages. Right: with range $r = 2$, the messages sent from node 0 only reach up to node 2, so forwards are necessary to reach the target, $n + 1$.

twice. Another option also used in practice is to cancel only if the sender of the duplicate message is closer to the destination than the current node.

3 The 1D analysis

In this section we study two fundamental scenarios in 1D, in which the leftmost of $n+2$ nodes sends k packets to the rightmost node (i.e., to a goal region which only contains the rightmost node). Each packet contains the position of its last (re)sender and its destination. Each node stores all received packets in a queue which is managed in one way or another depending on the protocol used. For simplicity, nodes are evenly spread at unit distance along the line. The n intermediate nodes form a dense bottleneck, a situation that can easily arise in practice. Once the transmissions start, collisions may happen. In order to cope with this problem, we assume fair medium access, i.e., the transmission is done by rounds, and in each round each node that has some packet to transmit has the same probability to transmit it.

Several aspects have to be taken into account when comparing the behavior of different protocols. The *success rate* measures the fraction of sent messages that actually reach the target. For those that arrive, the *hop count* indicates how many steps (forwards) are needed. In this paper we only focus on what we consider to be the most significant measure within this context, *RecMess*, which is defined as the maximum number of packets that a node receives. This parameter measures the work or energy consumption for a node, as well as the overall network load and therefore, its congestion.

Due to space limitation we only give the ideas of the proofs of the theorems.

3.1 Unbounded reach scenario

In the *unbounded reach scenario*, all nodes are within the range of each other. This setting recreates a rather frequent situation in which many messages must go through a high-density area.

Simple flooding. Under this protocol, all the nodes will receive and retransmit all the packets: $\textbf{RecMess} = nk$.

M heuristic. By definition, every node receives every message at most M times: $\textbf{RecMess} = Mk$.

T heuristic. Due to the unit-distance distribution of the nodes and assuming $T \in \mathbb{N}$, when a node transmits a packet, then this packet is deleted from the queues of its T left and its T right neighbors. Thus: $\lceil \frac{n}{2T} \rceil k \leq \textbf{RecMess} \leq \lceil \frac{n}{T} \rceil k$.

CD and CD-P heuristic. We note that **RecMess** in CD heuristic is higher than in CD-P heuristic. And move to analyzing the CD-P heuristic. We prove the following theorem.

Theorem 1 *In the unbounded reach scenario under the CD-P heuristic,* $\textbf{RecMess} = O(k \log n)$.

Proof idea. To prove this theorem we represent the message queues of the nodes as columns in a $k \times n$ table, and introduce k random variables that represent the rows' lengths. We bound the size of the longest row after every message transmission using probabilistic analysis of the kth order statistic.

Delay-based. We assume that the delay is chosen such that it increases by exactly one time step per node; that is, MD $= r$. The nodes delete messages from their queues when they receive them for the second time. Thus, every message is retransmitted only once, no matter which delay function is used. Therefore, $\textbf{RecMess} = k$.

In the variant in which the nodes delete messages from their queues only when they receive a duplicate from a node that is closer to the destination, we get:

$$\textbf{RecMess} = \begin{cases} 2^k & \text{if } k < \log n\,, \\ n + n(k - \log n) & \text{if } k \geq \log n\,. \end{cases}$$

3.2 Bounded reach scenario

In the *bounded reach scenario* each node can communicate with r neighbors to its left and r to its right, for some parameter r. This scenario generalizes the previous one, allowing to evaluate the effect that node density (indirectly related to r) has on the different protocols.

Lower bound. Any heuristic will need at least $\frac{kn}{r}$ retransmissions for all messages to reach the destination, as a message cannot progress by more than r nodes at a time. Every node receives a fraction $\Theta(\frac{r}{n})$ of all the messages, therefore, $\textbf{RecMess} = \Omega(k)$.

Simple flooding. Under this protocol, every node will receive each message from at most $2r$ of its neighbors: **RecMess** $= O(rk)$.

M heuristic. If $2r \leq M$ this protocol is equivalent to the previous one. If $2r > M$ it is equivalent to the M heuristic for the unbounded reach scenario. Therefore **RecMess** $= O(\min(M, 2r)k)$.

T heuristic. If $T \geq r$, no packet will ever be forwarded. If $T < r$, then each node u can receive a packet from at most $2r$ nodes. Each time it receives one, at least T and at most $2T$ of the nodes within reach of u delete the packet from their queues. Thus: **RecMess** $= O(\frac{rk}{T})$.

CD and CD-P heuristics. Consider a $n \times k$ table, where columns represent message queues of the nodes, and rows represent the messages that are in the queues. All messages start in the leftmost r columns. Whenever a node retransmits a message, it gets deleted from all nodes to its left and added to the $r-1$ nodes to its right. Thus, each message is always present in exactly r consecutive nodes (except at the end of the process).

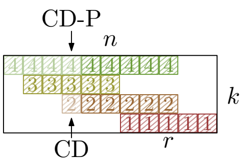

Figure 2: CD vs CD-P in bounded reach scenario.

When a node gets to retransmit, it picks the message with the lowest ID from its queue in CD heuristic, and with the highest ID in CD-P heuristic. Figure 2 illustrates the difference. On average, the progress a message makes to the destination when it gets retransmitted under the CD heuristic is smaller than under the CD-P. Thus, **RecMess** of CD heuristic is again greater than that of CD-P heuristic.

Theorem 2 *In the bounded reach scenario under the CD heuristic,* ***RecMess*** $= O(k^{3/2})$.

Proof idea. To prove this theorem we show that the expected progress a message makes when it is retransmitted is greater than $\frac{r}{\sqrt{k}+1}$, and the bound on the **RecMess** follows.

In contrast, the CD-P heuristic is optimal up to a constant factor with respect to **RecMess**.

Theorem 3 *In the bounded reach scenario under the CD-P heuristic,* ***RecMess*** $= \Theta(k)$.

Proof idea. We observe that the average progress a message makes on its way to the destination at each retransmission is greater than $\frac{r}{2}$. From which we deduce that **RecMess** $= O(k)$.

Delay-based. We discuss the case where delay_1 is used. The other functions can be analyzed in a similar way. Since all messages get deleted from its queue when heard by a node for the second time, **RecMess** $= O(\frac{nk}{r})$.

4 Concluding remarks

Beaconless geocast protocols are used in practice in 2D scenarios. They differ from the 1D ones in a few but important characteristics: the destination of a packet is defined as a region where more than one node may happen to be located; obstacles (like buildings, which cannot be traversed by the transmission signal) need to be surrounded, and local optimization strategies fail to guarantee delivery. Therefore, combinations of different strategies need to be used in order to achieve delivery guarantees and, at the same time, keep the network load within reasonable bounds. The network load analysis in this cases is difficult, and almost only experimental results exist. This is why we have started studying the 1D case. It has shown to be less trivial than we expected. Indeed, all protocols give rise to different load bounds, the CD and the CD-P heuristics being particularly tricky to analyze.

Acknowledgments I.K. and M.L. were supported by the Netherlands Organisation for Scientific Research (NWO) under projects no. 639.023.208 and 639.021.123 respectively. V.S. and R.S. were partially supported by projects MINECO MTM2015-63791-R and Gen. Cat. DGR 2014SGR46. R.S. was also supported by MINECO through the Ramón y Cajal program.

References

[1] R. J. Hall. An improved geocast for mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 10(2):254–266, 2011.

[2] M. Heissenbüttel, T. Braun, T. Bernoulli, and M. Wälchli. BLR: Beacon-less routing algorithm for mobile ad-hoc networks. *Computer Communications*, 27(11):1076–1086, 2004.

[3] C. Maihöfer. A survey of geocast routing protocols. *IEEE Communications Surveys and Tutorials*, 6(1-4):32–42, 2004.

[4] S. Rührup. Theory and practice of geographic routing. In *Ad Hoc and Sensor Wireless Networks: Architectures, Algorithms and Protocols*. Bentham Science, 2009.

[5] I. Stojmenovic and X. Lin. Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks. *IEEE Trans. Parallel Distrib. Syst.*, 12(10):1023–1032, 2001.

[6] M. Zorzi. A new contention-based mac protocol for geographic forwarding in ad hoc and sensor networks. In *IEEE Int. Conference on Communications (ICC)*, pages 3481–3485, 2004.