Breadcrumb Routing: Query-Response Geocast for Mobile Originators in Vehicular Networks

Julian Timpner, Mario Wozenilek and Lars Wolf
Institute of Operating Systems and Computer Networks
Technische Universität Braunschweig
Braunschweig, Germany
Email: (timpner|wozenil|wolf)@ibr.cs.tu-bs.de

Abstract—**Vehicular ad hoc networks allow connected vehicles to exchange and share information in order to improve traffic efficiency and safety as well as to provide infotainment functions. In order to cope with the challenging network conditions of the vehicular domain (e.g., intermittent connectivity), a disruption-tolerant networking architecture is suitable. In contrast to push-based communications (e.g, cooperative awareness messages) used in many proposed vehicular applications, we focus on a query-response model for requesting sensor information. Of particular interest to us is the question how vehicular queries can be efficiently routed to a destination area in such challenged environments, and, more importantly, how the response can be routed back to a moving query originator. We therefore present Breadcrumb Geocast Routing (BGR), a novel disruption-tolerant georouting protocol based on a trail of breadcrumbs left by the query originator. As an overlay protocol, BGR can be used on top of different georouting mechanisms, providing an efficient means to address moving originators. We study the influence of parameters, e.g., breadcrumb size and distance, and prove the effectiveness of the proposed solution in an extensive experimental evaluation, showing that BGR achieves significantly higher delivery rates compared to traditional geocast approaches, while avoiding up to 97 % of the traffic overhead of Epidemic and PRoPHET routing.**

I. Introduction

The automotive industry is on the brink of equipping modern vehicles with ad hoc communication systems[1], providing a means to share context information and thus enable new types of driver assistance systems. So far, most research in this area has focused on a *proactive model* to provide such systems with useful information, e.g., by broadcasting awareness and notification messages. While this approach is reasonable for information generally interesting to a large set of vehicles, e.g., safety messages, other applications benefit from a *reactive model*. The latter allows requests for specific information that need not be disseminated to many vehicles as it would consume both too much bandwidth and processing power in the network, e.g., sharing image data for up-to-date street-view impressions. However, because of the limited communication range of wireless radios and the highly dynamic structure of Vehicular Ad Hoc Networks (VANETs), often there is no end-to-end path between any two network nodes willing to exchange data. Delay-/Disruption-Tolerant Networks (DTNs) overcome this intermittent connectivity with a store-carry-forward approach and therefore do not require stable links. In a DTN, messages are stored at a node as long as there is no next hop available for forwarding a message. The node carries the message along its way and forwards it as soon as a new connection for forwarding becomes available. Thus, cars can share information that is not time-critical like road closures, parking situation information, entertainment content or even air pollution data. A challenge that arises from the use of geocast for queries is the routing of the response back to the originator of the query. Especially in vehicular networks, where messages can potentially have long delays, for example because of missing connectivity, the originator is likely to have moved relatively far away from the location from where the query was started. Thus, sending the results to the initial location is not sufficient. Enlarging the area to where the response is sent increases the overhead, as more nodes have to be involved in forwarding the message. Additionally, since the responding nodes typically do not know about the exact movement of the originator, the area is most likely chosen too large or too small.

A. Contributions of this paper

In this work, BGR is proposed, an approach that combines the technologies of DTNs and geocast and provides a mechanism to not only route a query into a geographic destination area, but also to route the response back to the moving originator of the query, even if its current location is unknown to the respondents. For this purpose, originators leave a trail of floating content [1] areas, which we refer to as *breadcrumbs* and which can be used to efficiently forward the response to its destination. To the best of our knowledge, we are the first to facilitate the floating content approach, which we extend significantly, for routing purposes. Since BGR can be used on top of different georouting mechanisms, it adds an efficient means to address moving originators even to protocols formerly ignoring the problem. It is evaluated extensively in different traffic scenarios. BGR can be easily extended and is shown to significantly improve the delivery rates of traditional geocast approaches in an efficient way.

This is the author's post-print version. Definitive version published in VNC'14.
DOI: 10.1109/VNC.2014.7013308

[1] http://www.car-to-car.org/

B. Outline

The remainder of this paper is structured as follows. We begin in Section II by studying what other authors have achieved in this field and how it relates to our work. We define relevant concepts and notations in Section III. In Section IV, the proposed BGR protocol is presented. Our simulation model is introduced in Section V. Section VI presents our experimental evaluation. The paper concludes in Section VII.

II. Related Work

A plethora of routing protocols exists in the literature. In this paper, we therefore only consider research work focusing on the problem of georouting in the vehicular domain and, in particular, on the problem of routing messages to moving nodes.

A. Georouting

In a recent survey of geocast routing protocols for VANETs, Allal et al. [2] present geocast protocols addressing the difficulties of this domain, such as "high mobility, frequent changes in topology, high and frequently variable density, long lifetime of nodes and regular moving patterns" [2]. For instance, Cached Geocast [3] is applicable in vehicular networks, since it already includes the feature of storing messages until there is a new link to forward the message or the message becomes outdated. Another example is Abiding Geocast [4], which keeps a message alive in a defined region for a defined period of time. However, none of the approaches solves the problem of sending replies to moving nodes, since they require a static destination area. BGR can be used on top of arbitrary geocast protocols, adding this missing capability. Geographic Source Routing (GSR) [5] proposes a routing strategy for VANETs in city environments. Yet, in order to find a vehicle's position, it floods the network with a position request. Further, it requires digital map data for optimization purposes. BGR, in contrast, does neither flood the network, nor does it require digital maps. GeoVANET [6, 7] aims to provide a geocast protocol for query processing in vehicular networks. In the proposed protocol queries are disseminated via broadcast. Receiving vehicles then process the query and reply indirectly, i.e., the answer is not routed to the requester directly (whose position is unknown anyway) but to a rendezvous point, referred to as a *mailbox*. In order to route the response to the mailbox, a geocast routing strategy is used. Eventually, the initiator of the query retrieves the responses from the mailbox, either by physically driving to within its communication range or by using the same georouting algorithm as in the step before. The latter, however, requires the vehicle to stop and not to move while the request is pending, which defeats the purpose. Similar to our approach, GeoVANET addresses the problem of routing messages to moving nodes by introducing stationary mailboxes. Additionally, our approach does neither rely on infrastructure nodes nor statically placed mailboxes, which induce the problem of finding suitable locations while the underlying network changes constantly. Static mailboxes also have the drawback of possible overload at popular mailboxes and unbalanced load in general. Instead, the approach presented herein uses dynamic areas that autonomously provide the service for actively forwarding a message to the originator of a query. Another contrast to GeoVANET is that the query is not broadcasted in the network, but instead is directly routed into the relevant destination area. Weak State Routing for Delay Tolerant Networks (WSR-D) [8] regularly disseminates state information to direct neighbors and to nodes that are located in regions in the moving direction of the target. This weak state represents the confidence in a node's location and is used to route messages towards it. WSR-D, however, initially sends messages in a random direction, assuming that some intermediate node will have information about the destination. In BGR, on the contrary, messages are sent directly to the last known location of the destination node and state is only kept along its actual path. BGR uses strong, i.e., deterministic, state to route messages, but as WSR-D it does not require reliable refresh or update messages. Moreover, WSR-D assumes that each node knows the end-point of its path, whereas BGR does not.

B. Query Processing

FleaNet [9] provides a digital marketplace on vehicular networks. Queries are generated and opportunistically distributed by vehicles and roadside units. FleaNet relies on mobility-assisted query dissemination where the query originator broadcasts the query only to its one-hop neighbors. The receiving vehicles do not relay the message, but try to provide answers from local information. Hence, the query is only disseminated because of the query originator's motion. Consequently, the problem of routing query results back to the originating vehicle does not arise. However, it significantly reduces the applicability of the approach, since it is not general enough to process queries over further distances. Roadcast [10] offers a content sharing mechanism for VANETs. As in FleaNet, queries are only disseminated to one-hop neighbors, with the same consequences. VITP [11] is an application layer protocol based on HTTP with support for both push and pull communications. The fully distributed ad hoc protocol allows the aggregation of data and only delivering summarized results. VITP-enabled peers in the area of interest cooperatively resolve the query, and return a reply. However, in order to transport the query into its destination area and in turn, transport the response back, any geographic routing can be used. This is similar to BGR routing, which also extends existing georouting protocols. Yet, VITP stipulates to extend the area over which the reply should be broadcasted if the querying node moves too far from the location from where it started the query. BGR, on the contrary, solves this problem more efficiently. However, it adopts VITP's computation phase for organizing a single response message on the basis of information of multiple nodes inside a geographic area.

In comparison, BGR neither requires all nodes to keep routing tables, nor any kind of infrastructure support, nor knowledge about vehicle trajectories or destinations. It therefore provides a versatile and lightweight extension to other geographic routing protocols. As the underlying geocast mechanism is exchangeable, a significant advantage of our approach is that it directly benefits from better or improved georoutings.

III. Preliminaries

In this section, relevant concepts and notations are defined. In particular, we introduce the concept of *breadcrumbs*, which are based on the floating content [1] approach. Yet, we significantly extend the original idea by introducing cooperation between and management of multiple floating content areas.

A. Definitions

In this work, we consider vehicles (*nodes*) that are equipped with ad hoc communication technology in order to form a network via Vehicle-to-Vehicle (V2V) communications. Benefits include that no infrastructure installations are necessary and that V2V communications are free of charge as opposed to cellular networks. All nodes are assumed to be able to determine their geographic position. Nodes sending a query towards a geographic destination area are referred to as *originators*, while nodes receiving, processing, and replying to the query are *respondents*. The location from where a query is sent is referred to as *querying location*.

B. Design Goals

In this work, we propose the Breadcrumb Geocast Routing (BGR) protocol, combining the geocast approach with delay-tolerant networking and response destination tracking via a trail of what we call *breadcrumbs*. BGR is applicable in mobile environments, where participants need to receive information that is not time-critical and corresponds to a specific geographic destination area, such as road conditions (available parking spots, road works, congestion) and roadside services (restaurant menus, gas station prices, etc.). When a node requires such information, it sends a query into the area of interest. While the addressed nodes in the area process the request, the position of the originator changes continuously, complicating the delivery of the responses. BGR addresses this problem by a best-effort strategy to efficiently find the originator. While BGR can be used on top of arbitrary geocast protocols (as explained in Section IV-A), we focus on the approach presented in [6] for evaluating BGR due to space constraints. In this approach, vehicles keep a message as long as they are getting closer to the message destination. When the distance to the destination increases, they forward the message to a neighbor moving towards it.

C. Breadcrumbs

In existing geocast approaches [4, 11], if a query response is to be sent to a moving originator, the destination area has typically to be quite large to ensure that the originator is still located within that area when the response arrives. Especially in a delay-tolerant environment, however, this approach has several disadvantages, since it leads to a high overhead (if many nodes are involved) or high loss of response messages (if the destination area does not include the originator). In this work, the originator leaves a trail of *floating content* [1] areas to ensure that responses can be routed towards it, even if it has already moved arbitrarily far away from its querying location. In analogy to a famous fairy tail[2], we refer to these floating content areas as *breadcrumbs*, since they are used to keep track of the originator's location and facilitate efficient routing decisions. Technically, a breadcrumb is a set of copies of special messages with the same content, which are passed on to each node inside a well-defined and restricted geographic area. Hence, the content is kept alive in that area, independent from the actual nodes that are present there, until a certain condition is met or the message lifetime expires. Breadcrumbs are based on anchor zones, which are defined by an anchor point P, the anchor zone radius r, an availability threshold a and a Time to Live (TTL). Fig. 1 depicts the according zones and parameters. Let h denote the distance of a node N from anchor point P. The distance h determines the probability p_f that N forwards its floating content message (in the following referred to as breadcrumb message) to all its neighbors that did not already receive it (which can be efficiently tested using Bloom filters [12]):

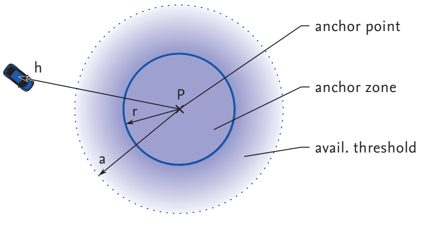


Fig. 1: Breadcrumb structure

$$p_f(h) = \begin{cases} 1 & \text{if } h \leq r \\ F(h) & \text{if } r < h \leq a \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where $F(h)$ with $\{h \in \mathbb{R} : r < h \leq a\}$ is defined as:

$$F(h) = \frac{h-r}{a-r}. \tag{2}$$

Consequently, the probability that a node deletes a breadcrumb message from its storage, $p_d(h)$, is:

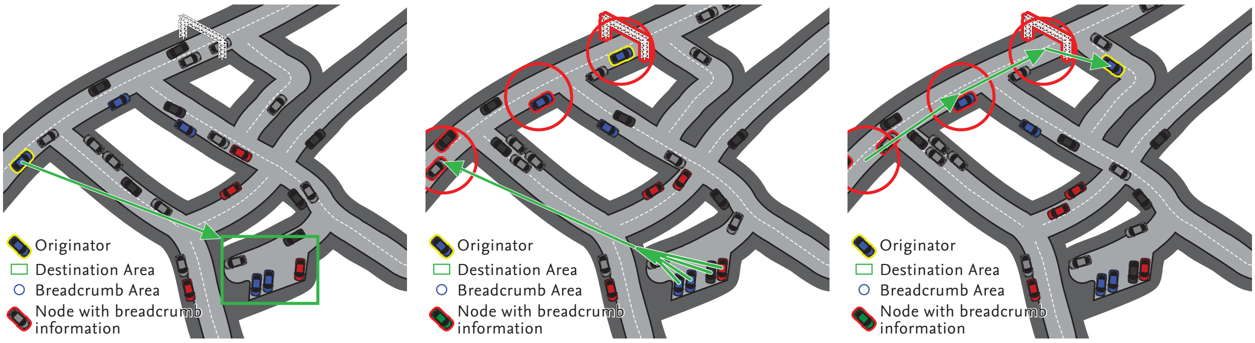
$$p_d(h) = 1 - p_f(h). \tag{3}$$

At minimum, breadcrumb messages contain information about the position of the preceding breadcrumb (if there is one), the position of the successive breadcrumb, the IDs of all queries the breadcrumb corresponds to and the anchor parameters P, r, a. Other content is possible, but, in general, the size of the breadcrumb messages should be minimized.

IV. Breadcrumb Geocast Routing

In this section, we present BGR, a geographic routing protocol that is not only able to send messages towards a destination area, but also to retrieve information from this area by efficiently routing the query responses back to the (moving) originator. The protocol is divided into the following three phases. In the *Query Phase*, the originator sends a query into the destination area. This phase ends when any node in the destination area has received the query. In the *Response Phase*, the response from a node inside the destination area (e.g., chosen by the VITP protocol) reaches the querying location. In the *Tracking Phase* the response is forwarded along a trail of breadcrumbs until it has reached the originator. If, at any point in time, the TTL of the query or the response expires, the corresponding phase ends, and no subsequent phase is entered. The three phases are shown in Fig. 2. During all phases,

[2] "Hansel and Gretel", published by The Brothers Grimm in 1812, leave a trail of breadcrumbs while walking in the forest in order to find their way back home.



(a) Query Phase

(b) Response Phase

(c) Tracking Phase

Fig. 2: BGR phases

each node broadcasts small 1-hop status messages, called beacons, for neighborhood discovery and location detection of neighboring nodes. In the following, the different phases are explained in detail.

A. *Query Phase*

When a vehicle requires information from a remote location, it sets up a query message $Q = \{R, T, P, D\}$, with:

- R being the actual request, such as “what is the current parking situation?”,
- T being the expiry time of the request, i.e., the TTL,
- P being the originator’s current position,
- D being the destination area.

Without loss of generality, we assume that the destination area is defined by a geographic center coordinate and a radius which determines the distance of the outline of the destination area to the center. The query Q is then sent to the destination area, using a traditional geocast. While we investigated several geocast algorithms for this purpose, in this work we are only describing results using GeoVANET [6] due to space constraints. However, the underlying geocast strategy can be easily exchanged.

Immediately after transmitting the message, the originator sets up a breadcrumb message m_i as described in Section III-C which will then be used to establish a breadcrumb i (i.e., a floating content area). Yet, the breadcrumb i is not *dropped* right away. Instead (and as depicted in Fig. 3), m_i is (1) stored while the originator (2) moves along its path until (3a) the breadcrumb distribution strategy triggers another breadcrumb message m_{i+1} to be set up. Now, (3b) the stored breadcrumb message m_i is updated with the position of the new breadcrumb $i + 1$ (which is to be dropped at the vehicle’s new current position). Because the originator already moved on and away from the anchor point of the old breadcrumb i, the stored breadcrumb message m_i is sent to its anchor point using geocast and (4) breadcrumb i is established in the destination area. This process repeats as the originator moves along its path. The originator stops dropping breadcrumbs once it has received the response to its query or the maximum waiting time has been exceeded. In the latter case, the request is considered to be failed and the breadcrumb information is deleted. This delayed creation process has several advantages. First, it is ensured that a breadcrumb always has the correct information about its successor. Second, if the originator does not move away from the querying position, no breadcrumb is necessary and none will be dropped. Naturally, the question of where the next breadcrumb should be dropped arises. Since a problem similar to finding the optimal position was shown to be NP-hard [13], a detailed evaluation of dropping strategies is out of scope for this paper. Hence, for the purpose of this work, breadcrumbs are dropped periodically after the originator has covered a certain (parameterizable) distance from the last breadcrumb or the querying location. Depending on the inter-breadcrumb distance, two special cases are possible. If only one breadcrumb ($n = 1$) has been dropped, this strategy is similar to Mobile IP [14], where a home agent’s address (location in our case) is known and can therefore be used to send and forward messages to a mobile node. In the second special case, multiple spatially-overlapping breadcrumbs ($n \to \infty$) have been dropped. This effectively leads to a single extended geocast region, which is similar to existing approaches that choose arbitrarily large destination areas to reach nodes with unknown locations [4, 11]. Yet, in comparison the breadcrumb approach would still be more efficient, since the destination area would be elongated along the actual vehicle path, as opposed to a circular extension in all directions. Anyway, we expect the general case in which the number n of breadcrumbs is $1 < n \ll \infty$ to be more reliable and efficient, since a smaller number of nodes is involved in the transmission of a response. Hence, we focus on this case.

B. *Response Phase*

The *Response Phase* starts when the message has reached (the first node in) the destination area. Inside the destination area, the nodes respond to the message and merge their results into one message that contains all answers to the initial request. This can be achieved by adopting VITP [11], in which the response message is sent once a return condition is satisfied. It is sent (following the same geocast procedure as in the Query Phase) to the initial breadcrumb at the querying location.

C. *Tracking Phase*

Once a response has reached any breadcrumb that holds information about the originator of the corresponding query, the Tracking Phase begins. Any breadcrumb node receiving

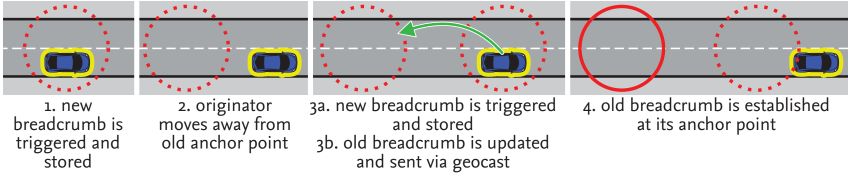

1. new breadcrumb is triggered and stored
2. originator moves away from old anchor point
3a. new breadcrumb is triggered and stored
3b. old breadcrumb is updated and sent via geocast
4. old breadcrumb is established at its anchor point

Fig. 3: Creation and delayed dropping process of breadcrumbs

such a response message will replace the destination area of the response message with the position of the successive breadcrumb. The node will use a random backoff timer before trying to forward the response to the next breadcrumb, minimizing the number of nodes forwarding the same message. Accordingly, the response is forwarded from a breadcrumb to its successor. This forwarding continues along the trail of breadcrumbs, until the originator has received the response, as depicted in Fig. 2c. After a node in a breadcrumb area has forwarded the response, the nodes of that particular breadcrumb can delete both the response and the tracking information. Thus, the breadcrumb disappears after forwarding the response (or at the latest, when the breadcrumb's TTL is exceeded). It is worth noting that it is not always necessary to pass through the entire chain of breadcrumbs, as each breadcrumb does not depend on its predecessor, but all successors. Consequently, whenever breadcrumb i receives a corresponding response, it will forward the message to breadcrumb $i+1$; breadcrumbs $< i$ will not be involved. Thus, a certain degree of resilience against a disconnected breadcrumb trail is introduced. Even in the worst case, though, BGR gracefully degrades back to the underlying georouting protocol and does thus not perform worse. Evaluating further mechanisms such as repairing or shortening a disconnected trail is left for future work.

V. Simulation Setup and Scenarios

In order to evaluate BGR, simulations with the widely-used The Opportunistic Network Environment (The ONE) [15] simulator have been conducted. In this section, we present the used scenarios and models.

A. Simulation Scenarios

The nodes move with a random map-based mobility model. For the goal of this paper, the following two scenarios have been used.

1) San Francisco: The San Francisco, CA scenario[3] covers the area around the (Lower) Pacific Heights neighborhood and Marina District. The area is approximately 5.39 km^2 in size and includes 108.78 km of roads. As is common in U.S. cities, the structure of the road network is for the most part rectangular shaped with two parks in the center of the scenario. Fig. 4a shows the layout of the used map.

2) Helsinki: For the second scenario, the downtown area around the city center of Helsinki, Finland is used. The area is 14.67 km^2 in size and includes 119.2 km of Helsinki's road network. As in the San Francisco scenario, all roads are assumed to be bidirectional. Fig. 4b shows the layout of the simulated road network, based on [15].

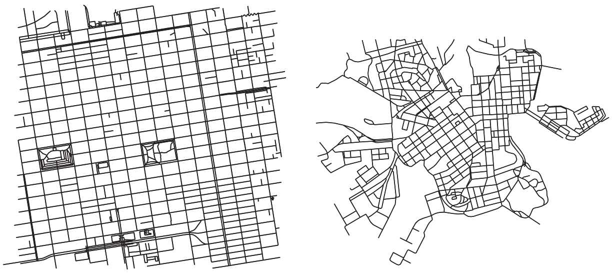

(a) San Francisco (b) Helsinki

Fig. 4: Road networks

B. Simulation Setup

If not stated otherwise, the default values shown in Table I are used for the simulation. To ensure comparability to conventional geocast routing algorithms, that do not consider a response to a specific request, and to avoid effects that are caused by parts of the protocol other than the routing decisions and mechanisms, only the routing of responses to the requesting car is investigated. Therefore, the queries themselves are immediately and always successfully transmitted into their destination area, and one respondent has already been elected (using VITP) to forward the response to the originator. However, in order to take the transmission and the processing time of the query into account, a random "response delay" is applied, before a response is initiated. Without loss of generality, this reduces the simulation's overhead and allows it to focus on the evaluation of the breadcrumb approach for routing responses to a moving originator. The inter-breadcrumb distance d defines how far the originator has to move away from the last breadcrumb position before a new breadcrumb is created. Therefore, the parameter determines how many breadcrumbs are created. The breadcrumb size $s = a$ includes the anchor zone and the availability threshold (see Section III-C). Further, only currently driving cars initiate queries (as opposed to parking vehicles). Thereby, each newly created query causes breadcrumbs to be dropped by the originator. Messages are created at random nodes from the scenario area with a random waiting time before each new message creation. The average distance between originator and destination area in the San Francisco and Helsinki scenario are 1200 m and 1500 m, respectively. To ensure comparability between the different scenarios, the average number of vehicles per road kilometer is 17 veh/km, which is based on real-world traffic data [16] of Braunschweig, Germany during morning rush hour. Since the simulator is only able to specify the range of 100 % successful communications, the assumption of a rather conservative transmission range of 25 m is made. Based on the urban environment of the scenarios, prior research [17] on the communication distance of IEEE 802.11p found the probability

[3] http://www.tm.kit.edu/~mayer/osm2wkt/

TABLE I: Default settings for simulations

Setting	Value
Breadcrumb distance d	100 m
Breadcrumb size s	50 m
Time between new messages	120 to 360 s
Response delay	480 s
Randomly moving cars	17 veh/km[1]
Simulated period	12 h
Transmission rate	22 Mbit/s
Transmission range	25 m
Time to live (Request)	60 min
Driving speed	18 to 54 km/h

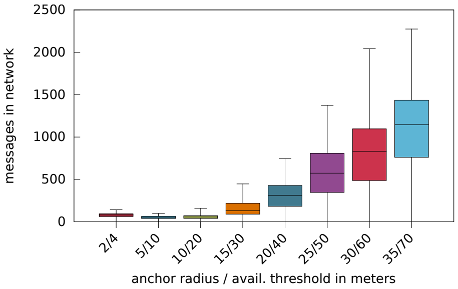


Fig. 5: Overall network traffic in Helsinki

of successful communication in this range and environment to be sufficiently high, i.e., above 85 %.

VI. Experimental Evaluation

In this section, we first explore the impact of breadcrumb parameters on the overall performance in order to find good default values. We then present a comparative performance analysis of BGR with other widely-used routing protocols. Due to space constraints, we can only present selected results and have to leave others (e.g., effect of TTL on query success ratio and propagation delay) for an extended version.

A. Breadcrumb Distribution

The combination of the breadcrumb size s and the inter-breadcrumb distance d (the distance between anchor points of subsequent breadcrumbs) defines the breadcrumb distribution type. Both parameters were evaluated separately and in combination to understand their influence on the routing and to find a satisfying trade-off between cost and benefit.

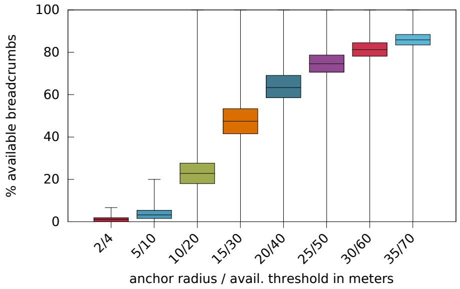


Fig. 6: Available breadcrumbs in Helsinki ($d = 100\,\mathrm{m}$)

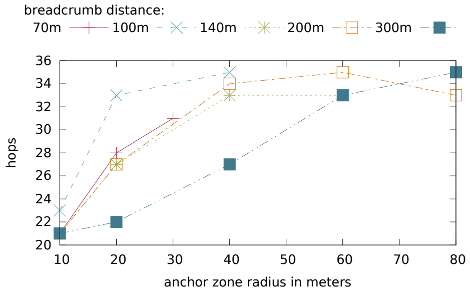


Fig. 7: Median hop count

1) Breadcrumb size: The breadcrumbs' size s depends on two parameters—the radius r of the anchor zone and the availability threshold a. In [1], the authors already discovered that r has a higher impact on the performance than a, which is why we set $a = 2r$, ergo $s = a = 2r$. With an increasing breadcrumb size, the overall network traffic increases, as shown in Fig. 5. This result is clearly expected, since larger sizes imply that more vehicles are in the corresponding zones and therefore exchange and store breadcrumb messages. Although it was assumed in [1] that $r \ll 200\,\mathrm{m}$ will not lead to sufficient replication, Fig. 8b shows that smaller breadcrumbs do make sense if not used separately but in a cooperative context. Fig. 6 shows that even with $r = 30\,\mathrm{m}$, about 80 % of the breadcrumbs are available at their anchor point, i.e., the breadcrumb messages are kept and forwarded by vehicles in the anchor zone. For smaller sizes, however, the number of available breadcrumbs decreases rapidly. The correlation of available breadcrumbs and delivered messages becomes evident by comparing Fig. 6 and Fig. 8b. As the overall network traffic should be kept low, smaller breadcrumb sizes are preferable. Yet, Fig. 6 and Fig. 8b indicate that $r < 10\,\mathrm{m}$ leads to almost no breadcrumbs being available, thus degrading BGR to the underlying geocast routing. It is also worth noting, that the delivery rate exceeds 90 % with $r \geq 20\,\mathrm{m}$ in the San Francisco scenario, where the geometric structure of the road network seems to increase the effectivity of the available breadcrumbs, as opposed to the Helsinki scenario. On the other hand, the increase in delivered messages stagnates for larger r, as the breadcrumbs start to overlap because of their size in combination with the breadcrumb distance of 100 m. Since the overhead, however, increases for larger radii, $r = 25\,\mathrm{m}$ seems to be a reasonable trade-off between delivery rate and network traffic. The hop count is strongly related to the number of available breadcrumbs, as Fig. 6 and Fig. 7 show. As expected, the more breadcrumbs are available the more redirections are observable and, ergo, the median hop count increases.

2) Distance-Size-Relation: The other parameter that influences the distribution of the breadcrumbs is the distance d between subsequent breadcrumbs' anchor points. As Fig. 8a shows, the delivery rate is negatively affected by increasing d. Yet, it becomes clear that the size of a breadcrumb has a larger impact on the performance of BGR than the distance, since the difference in the delivery rate depending on d is much less than the difference depending on r. However, to yield a high delivery rate, breadcrumbs should be close to each other (without overlapping, to avoid useless traffic). Consequently, to ensure high delivery rates, the distance d should satisfy $d \approx 2a$. Given $r = 25$ as explained in Section VI-A1, a distance $d = 2a = 2(50\,\mathrm{m}) = 100\,\mathrm{m}$ is reasonable.

B. Comparative Performance Analysis

In the following, BGR is compared to the underlying GeoVANET protocol, which serves as an instance of GeoRouting protocols that do not address the problem of moving destination nodes. Epidemic Routing [18] is used as a common baseline, due to the fact that it provides us with the theoretical upper bound in terms of delivery rate, and the lower bounds in terms of delivery time and hop count. Since this is common practice [19, 20], BGR can in turn be easily compared to other approaches as well. An additional comparison to the PRoPHET [21] and Spray and Wait [20] protocols complements the evaluation.

a) Network Traffic: Fig. 9 shows the median number of messages in the network for the different protocols. For BGR, this includes all copies of breadcrumb messages, while Spray and Wait is used in binary mode and the maximum number of copies is set to 5. On the one hand, it is observable that BGR causes significantly less network traffic than both Epidemic and PRoPHET. To be more precise, it reduces the number of messages in the network by up to 97.6 % and 97.5 %, respectively. On the other hand, BGR produces more messages than regular geocast without the breadcrumb approach, since breadcrumb messages are replicated in their respective areas. Since the number of copies for Spray and Wait is limited to 5 messages, it has a very low traffic overhead. In turn, its delivery rate and propagation delay are significantly worse than BGR, as described in the following.

b) Delivery Rate: Fig. 10 shows the delivery rate per scenario. Epidemic and PRoPHET achieve 100 % delivered messages in both scenarios, due to the huge number of copies in the network. Delivery rates for Spray and Wait ($n = 5$) range between 45 % to 63 %. GeoRouting shows the lowest delivery rate, since it is not able to find the moving destination node on purpose, as wouldn't most other geocast approaches. Instead, GeoRouting has to rely on coincidence to deliver the message. BGR outperforms GeoRouting significantly in both scenarios, achieving almost 100 % compared to 31 % in the SF scenario. In the Helsinki scenario, BGR achieves 79 % compared to 29 % with GeoRouting. As explained in Section VI-A, the street layout in Helsinki negatively affects the distribution of breadcrumbs, resulting in a lower delivery rate than in the SF scenario. These results show that the breadcrumb approach works as intended and significantly improves upon existing georouting schemes by boosting the delivery rate while avoiding the overhead of protocols that are able to find moving destinations, such as Epidemic and PRoPHET.

c) Hop Count: Fig. 11 shows the median hop counts per message. As expected, BGR adds several hops to the performance of GeoRouting because of the (intended) redirection of messages along the trail of breadcrumbs. Since GeoRouting itself already requires almost twice as many hops as Epidemic or PRoPHET (which is also acknowledged as a drawback by its authors [6]), BGR naturally "inherits" the high hop count. Since the underlying geocast algorithm is exchangeable, though, BGR will directly benefit from an improved approach. Spray and Wait requires the least hops, since the number of copies is limited and nodes must transport messages by locomotion instead of transmission. In terms of propagation delay, BGR is almost on par with GeoRouting in the San Francisco scenario, while the additional hops in Helsinki lead to a slightly higher delay.

VII. Conclusion

In this paper, we present the Breadcrumb Geocast Routing (BGR), a georouting protocol for vehicular networks. BGR not only enables network nodes to route queries into a geographic area, but to route the response back to a moving originator, for which only the initial querying location is known. For this purpose, breadcrumbs are introduced which are a set of messages dynamically kept available for a defined period of time at specific locations along the path of the originator. These breadcrumbs sequentially forward a query response towards the originator. The protocol was evaluated in two different urban scenarios using The ONE simulator. In the evaluation, the performance of BGR was compared to reference routing protocols such as Epidemic, PRoPHET, Spray and Wait and traditional geocast. Additionally, the influence of different parameters such as breadcrumb distribution was evaluated. The results show that BGR avoids up to 97 % of the traffic overhead of Epidemic and PRoPHET, while increasing the delivery rate of an existing georouting protocol significantly from about 31 % to almost 100 %. The breadcrumb size has the greatest impact on the performance of BGR, making it a key parameter for the trade-off between network traffic and delivery probability. As future work, we plan to study optimizations, e.g., sending responses to multiple breadcrumbs simultaneously, to reduce the hop count and to provide shorter delays. Further, we are going to investigate how breadcrumbs can be dynamically adapted to the street layout, node density and other scenario-specific parameters.

Acknowledgment

This project has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under Grant Agreement Number [269916, V-Charge].

References

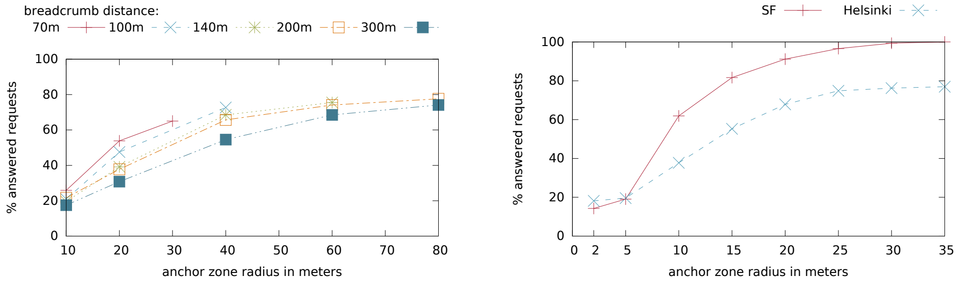
[1] J. Ott et al. "Floating Content: Information Sharing in Urban Areas". In: *Proc. IEEE Int. Conf. on Pervasive Computing and Communications (PerCom).* Seattle, WA: IEEE, Mar. 2011, pp. 136–146.

[2] S. Allal and S. Boudjit. "Geocast Routing Protocols for VANETs: Survey and Guidelines". In: *Proc. Sixth Int. Conf. on Innov. Mobile and Internet Services in Ubiq. Comp.* Palermo, Italy: IEEE, July 2012, pp. 323–328.

[3] C. Maihofer and R. Eberhardt. "Geocast in Vehicular Environments: Caching and Transmission Range Control for Improved Efficiency". In: *Proc. IEEE Intelligent Vehicles Symposium (IV '04).* Parma, Italy: IEEE, June 2004, pp. 951–956.

[4] C. Maihöfer, T. Leinmüller, and E. Schoch. "Abiding Geocast: Time-stable Geocast for Ad Hoc Networks". In: *Proc. 2nd ACM Int. Workshop on Vehicular Ad Hoc Networks (VANET '05).* Cologne, Germany: ACM Press, Sept. 2005, pp. 20–29.

[5] C. Lochert et al. "A Routing Strategy for Vehicular Ad Hoc Networks in City Environments". In: *Proc. IEEE Intelligent Vehicles Symposium (IV '03).* Columbus, OH: IEEE, June 2003, pp. 156–161.



(a) Delivery rates in the Helsinki scenario

(b) Delivery rates ($d = 100\,\mathrm{m}$)

Fig. 8: Influence of inter-breadcrumb distance and breadcrumb size on delivery rate

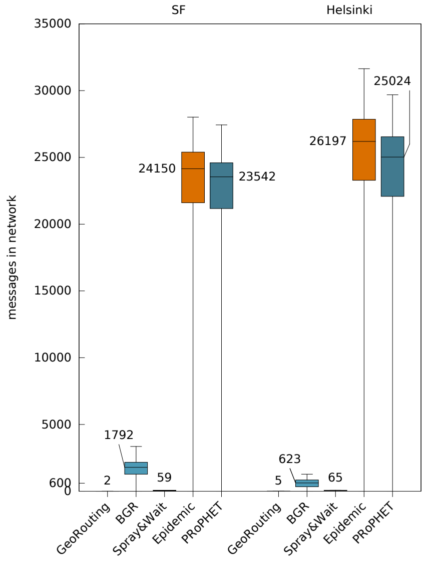


Fig. 9: Network traffic comparison

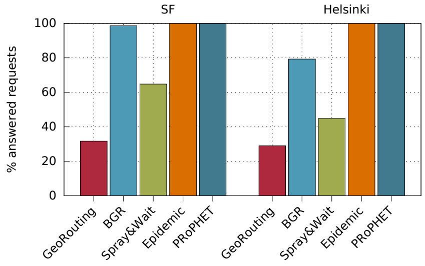


Fig. 10: Delivery rate comparison

[6] T. Delot et al. “Decentralized Pull-Based Information Gathering in Vehicular Networks Using GeoVanet”. In: *Proc. 12th IEEE Int. Conf. on Mobile Data Management (MDM ’11)*. Vol. 1. Lulea, Sweden: IEEE, June 2011, pp. 174–183.

[7] T. Delot et al. “GeoVanet: A Routing Protocol for Query Processing in Vehicular Networks”. In: *Mobile Information Systems* 7.4 (2011), pp. 329–359.

[8] U. G. Acer, S. Kalyanaraman, and A. A. Abouzeid. “DTN routing using explicit and probabilistic routing table states”. In: *Wireless Networks* 17.5 (May 2011), pp. 1305–1321.

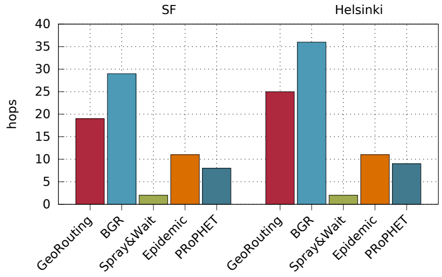


Fig. 11: Hop count comparison

[9] U. Lee et al. “FleaNet: A Virtual Market Place on Vehicular Networks”. In: *Proc. 3rd Int. Conf. on Mobile and Ubiquitous Systems*. San Jose, CA: IEEE, July 2006, pp. 358–365.

[10] Y. Zhang, J. Zhao, and G. Cao. “Roadcast: A Popularity Aware Content Sharing Scheme in VANETs”. In: *Proc. 29th IEEE Int. Conf. on Distributed Computing Systems*. Montréal, Canada: IEEE, June 2009, pp. 223–230.

[11] M. D. Dikaiakos et al. “VITP : An Information Transfer Protocol for Vehicular Computing”. In: *Proc. 2nd ACM Int. Workshop on Vehicular Ad Hoc Networks (VANET ’05)*. Cologne, Germany: ACM Press, Sept. 2005, pp. 30–39.

[12] M. Mitzenmacher. “Compressed Bloom Filters”. In: *IEEE/ACM Transactions on Networking* 10.5 (Oct. 2002), pp. 604–612.

[13] S. S. Chawathe. “Inter-Vehicle Data Dissemination in Sparse Equipped Traffic”. In: *Proc. IEEE Intelligent Transportation Systems Conf. (ITSC ’06)*. Toronto, Canada: IEEE, Sept. 2006, pp. 273–280.

[14] C. Perkins. *IP Mobility Support for IPv4, Revised*. RFC 5944 (Proposed Standard). Internet Engineering Task Force, Nov. 2010.

[15] A. Keränen, J. Ott, and T. Kärkkäinen. “The ONE Simulator for DTN Protocol Evaluation”. In: *Proc. 2nd Int. Conf. on Simulation Tools and Techniques (Simutools ’09)*. Rome, Italy: ICST, Mar. 2009, 55:1–55:10.

[16] M. Caliskan, D. Graupner, and M. Mauve. “Decentralized Discovery of Free Parking Places”. In: *Proc. 3rd Int. Workshop on Vehicular Ad hoc Networks (VANET ’06)*. Los Angeles, CA: ACM Press, Sept. 2006, pp. 30–39.

[17] L. Stibor, Y. Zang, and H.-J. Reumerman. “Evaluation of Communication Distance of Broadcast Messages in a Vehicular Ad-Hoc Network Using IEEE 802.11p”. In: *Proc. IEEE Wireless Communications and Networking Conf.* Kowloon, China: IEEE, Mar. 2007, pp. 254–257.

[18] A. Vahdat and D. Becker. *Epidemic Routing for Partially-Connected Ad Hoc Networks*. Tech. rep. Duke University, 2000, pp. 1–14.

[19] J. Link, D. Schmitz, and K. Wehrle. “GeoDTN: Geographic Routing in Disruption Tolerant Networks”. In: *Proc. IEEE Global Telecomm. Conf. (GLOBECOM ’11)*. Houston, TX: IEEE, Dec. 2011, pp. 2432–2436.

[20] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. “Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks”. In: *Proc. ACM SIGCOMM Workshop on Delay-tolerant Networking (WDTN ’05)*. Philadelphia, PA: ACM Press, Aug. 2005, pp. 252–259.

[21] A. Lindgren et al. *Probabilistic Routing Protocol for Intermittently Connected Networks*. RFC 6693 (Experimental). Internet Engineering Task Force, Aug. 2012.