Maintenance and Privacy in Unstructured GeoCast Overlays for Smart Traffic Applications

Bernhard Heep and Ingmar Baumgart
Institute of Telematics, Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany
Email: {heep,baumgart}@kit.edu

I. Introduction

In times of increasing mobility and climate change, there is a need for new services to cope with the special challenges of electric vehicles, like recuperation and charging management. Established systems of so called *smart traffic applications* are usually server-based and bear the risk of uncontrollable gathering of private data by service providers. In this paper we propose a decentralized overlay protocol for smart traffic applications that meets the requirements of several scenarios of future traffic. Our system offers a scalable *GeoCast* [1] service, where participants (e.g. vehicles) are able to gain information from specific geographic regions. In the following, we briefly describe three scenarios that would benefit from such a GeoCast service. After that, the main part of this paper discusses maintenance and privacy issues regarding the GeoCast overlay.

II. Application Scenarios

The following scenarios are considered as potential smart traffic applications using our GeoCast overlay:

- **Dynamic and cooperative automotive navigation**: Traffic participants could exchange information about traffic jams and closed roads as well as their average speed in a decentralized manner. Using this information, drivers can adjust their precalculated route if necessary.
- **Discovery and reservation of charging stations**: Due to the fact that electric cars still have very limited range and the network of appropriate charging stations is sparse, a GeoCast service could enable the discovery and reservation of free charging stations in regions that lie ahead in the vehicles' directions of travel.
- **Carsharing and carpooling management**: Participants of carsharing and carpooling communities could gain information about free and fully charged vehicles at nearby parking sites or regions.

In all these scenarios, vehicles need specific information from particular geographic regions. A generic and decentralized GeoCast service would meet this requirement. Opposed to centralized services like *Google Live Traffic* [2], it additionally facilitates user privacy by enabling data avoidance and data minimization.

III. Overlay-based GeoCast Service

Our generic GeoCast approach is based on overlay techniques, since overlay-based systems are scalable, self-organizing, and independent of the utilized network access technologies used by the participants. In particular, the overlay structure we propose is based on the unstructured *Gia* [3] overlay extended by a GPS-based neighbor selection mechanism. Similar to *GeoKad* [4], each overlay participant maintains n_c concentric circles of increasing sizes as neighborhood table. However, we additionally consider the direction of travel and the areas of interest of participants, when choosing overlay neighbors. The local node's GPS coordinates determine the center of the circles. Each circle holds a maximum of n_r neighbor nodes whose coordinates point into the circle (Fig. 1).

Basically, the GeoCast service offers a simple API for participating nodes. Its main function `geoCast()` is utilized to send messages into geographic regions. The shapes and sizes of these regions are specified by given parameters. At each node, messages are recursively forwarded to neighbors that are positioned closer to the destination region. Within that region, messages are optionally flooded. Alternatively, nodes in the destination region are identified by performing lookup procedures comparable to *iterative lookups* in structured P2P overlays.

For example, in the *dynamic and cooperative automotive navigation* scenario, nodes send messages into regions that will be passed when following precalculated routes. Nodes that are currently positioned in these regions receive the messages and respond if they have the requested information (*PULL mode*). In the alternative *PUBLISH/SUBSCRIBE mode*, inquired nodes do not answer directly, but only in case the state of traffic changes. To contact other vehicles without request (e.g. to warn following vehicles about an emergency situation), the *PUSH mode* is used.

To join the GeoCast overlay, a node sends a *JOIN_REQUEST* message to an arbitrary *bootstrap node*. The message's destination is the node's current geographic position. Overlay nodes around these GPS coordinates that receive the message respond with lists of potential neighbor nodes for the new node's neighborhood table.

A. Maintaining the Overlay

To maintain the overlay structure, i.e. to keep the overlay graph connected, each node has to ensure that it knows other nodes that are positioned in all geographic directions from the local node's point of view (determined by a *covering satisfaction function* f_{cover}). For this, each node periodically sends *DISCOVER* messages with a period $t_{discover}$ into geographic regions that are not sufficiently covered by the nodes currently listed in the node's neighborhood table.

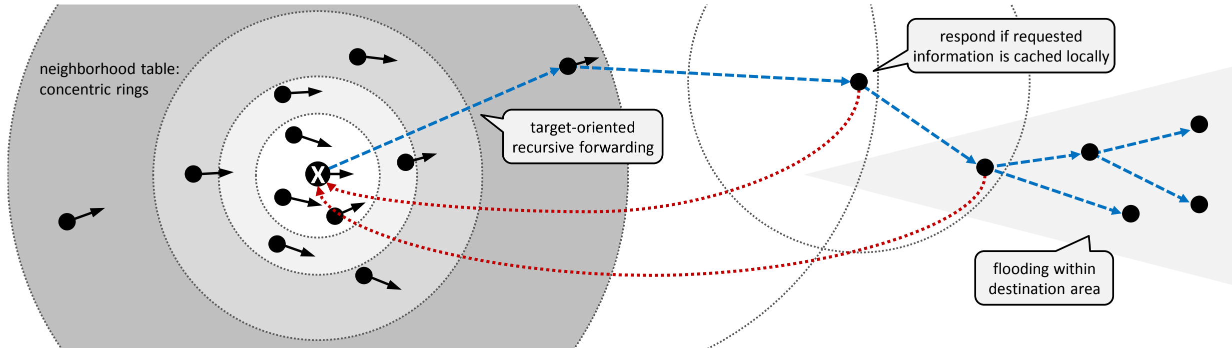

Fig. 1. Concentric circles as node X's neighborhood table (with $n_c = 4$, $n_r = 3$) and recursive forwarding of messages into destination area.

In applications scenarios where nodes are only interested in information located ahead in their direction of travel (e.g. the state of traffic in a distance of 100 km on the same highway) neighbor selection can be adjusted in a way that only mobile nodes with a similar direction of travel are considered as potential neighbors.

After a message is forwarded into the destination region, responding nodes can be added to an extra table of n_a so called *anchor nodes* by the query initiator. Forwarding messages directly to these nodes will speed up future queries into the same region. Supposed that they have the same direction of travel as the querying node, anchor nodes might also have useful information in the future. For further minimization of traffic and faster gaining of information, nodes *cache* all information they get for t_{cache} seconds. This accelerates the retrieval of information for e.g. following vehicles on a highway.

B. Privacy Considerations

Today's systems for dynamic automotive navigation collect all available data about the current state of traffic at a provider's central server. This information is either periodically broadcasted into a large region (e.g. via a radio traffic channel) to all vehicles which then dynamically calculate their optimal routes locally. However, the data distributed this way often lacks of currentness. Another approach is that dynamic routes are calculated at a provider's central server that gathers information from all participants' GPS devices (like e.g. *Google Live Traffic*). This approach bears the risk that providers create user profiles thus violate privacy demands.

In contrast, our overlay-based approach facilitates data minimization and data avoidance, i.e. requests and responses are both exchanged in a limited geographic area only. Due to the fact that only participants are involved in the exchange of data, creating user profiles by a provider is impossible. Vehicles are identified by both the IP address and a nodeId. While IP addresses change every time the node switches its access network, nodeIds are independent of the utilized network access technology. NodeIds are calculated by hashing the public keys corresponding to certificates issued by a central *Certificate Authority* (CA). All sent messages have to be signed using the private key bound to this certificate. To retain privacy, these certificates represent only temporarily valid pseudonyms. Moreover, the CA does not receive any location data, it only registers users, issues and exchanges certificates, and receives complains about malicious participants. This way, participants only know the temporary IPs and nodeIds as well as the current geographic positions of other nodes as long as the corresponding vehicles are still part of the overlay. When starting a new trip, IP, nodeId, and position may have changed thus privacy is ensured.

To avoid the recognition of vehicles that rejoin the overlay, nodes must choose their bootstrap node randomly. To impede a mapping of vehicles in sight to nodes in the overlay, GPS coordinates are sent moderately falsified.

IV. Conclusion & Future Work

In this paper we presented a decentralized GeoCast service for smart traffic application, provided by a GPS-based overlay network. Due to its advantages regarding scalability, data minimization and data avoidance, our GeoCast service could be an alternative to established centralized systems. In this context, we discussed maintenance issues as well as privacy considerations. Our next steps will be the implementation and the evaluation of the proposed system using the OverSim [5] simulation framework. For this we plan to extend OverSim's API [6] to support GeoCast and by an underlay model based on cartographic material from `openstreetmap.org`.

References

[1] J. C. Navas and T. Imielinski, "Geocast - Geographic Addressing and Routing," in *Proceedings of the* 3^{rd} *Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '97)*. Budapest, Hungary: ACM Press, Sep. 1997, pp. 66–76.

[2] "Google live traffic / maps navigation - website," http://www.google.com/mobile/navigation/, 2012, [online] accessed on 03/15/2012.

[3] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, "Making gnutella-like P2P systems scalable," in *Proceedings of the annual Conference of the ACM Special Interest Group on Data Communication (ACM SIGCOMM '03) on the Applications, Technologies, Architectures, and Protocols for Computer Communications*. Karlsruhe, Germany: ACM Press, Aug. 2003, pp. 407–418.

[4] M. Picone, M. Amoretti, and F. Zanichelli, "GeoKad: A P2P Distributed Localization Protocol," in *Proceedings of the* 8^{th} *IEEE International Pervasive Computing and Communications Conference (PERCOM 2010) Workshops*. Mannheim, Germany: IEEE Computer Society, Mar. 2010, pp. 800–803.

[5] I. Baumgart, B. Heep, and S. Krause, "OverSim: A Flexible Overlay Network Simulation Framework," in *Proceedings of the* 10^{th} *IEEE Global Internet Symposium (GI '07) in conjunction with IEEE INFOCOM 2007*. Anchorage, AK, USA: IEEE Computer Society, May 2007, pp. 79–84.

[6] I. Baumgart, B. Heep, C. Hübsch, and A. Brocco, "OverArch: A common architecture for structured and unstructured overlay networks," in *Proceedings of the* 15^{th} *IEEE Global Internet Symposium in conjunction with IEEE INFOCOM 2012*. Orlando, FL, USA: IEEE Computer Society, Mar. 2012.