ROME: Road Monitoring and Alert System through Geocache

Bin Zan Tingting Sun Marco Gruteser Yanyong Zhang
WINLAB, Rutgers University
Technology Center of New Jersey
671 Route 1 South
North Brunswick, NJ 08902-3390
{zanb,sunting,gruteser,yyzhang}@winlab.rutgers.edu

***Abstract*—We present a road monitoring and alert system implemented using a novel Geocache concept to enable efficient spatial monitoring in a mobile distributed sensing scenario. Technology trends have led to the integration of positioning, communications, and sensing capabilities into mobile entities such as cars and cellular phones, enabling them to monitor and report on their surroundings. We consider scenarios where events of interest must be detected from aggregated readings of multiple devices. For example, road monitoring could infer road defects from increased vibrations or road hazards from repeated emergency braking at the same location. This raises the challenge of aggregating sensor data from multiple mobiles that have passed the same location with efficient usage of communication resources. We introduce the Geocache concept, which allows anchoring sensor information at specific spatial coordinates, rather than storing it in a designated node. The Geocache protocol in either its relayed or delayed variant will then opportunistically determine a storage vehicle near the Geocache and hand the data off as vehicles pass by. We show through simulations with synthetic and realistic automotive position traces that relayed Geocache reduce messaging overhead by 66% compared to a baseline periodic broadcasting scheme and a Boomerang Geocache can provide a further 71% reduction when only few of the passing cars are able to sense the event.**

I. Introduction

The increasing integration of sensors and wireless communication devices into highly mobile platforms such as automobiles is enabling novel pervasive monitoring services that continuously sense the surrounding environment and report events of interest on a real-time basis. Today's higher-end cars already carry a range of sensors, e.g., rain gauges, accelerometers, GPS, wheel rotation/traction sensors, and cameras, which can be used to report on a variety of road conditions such as potholes, obstacles, or slipper driveways. The ability to timely communicate such events with road management authorities which can issue warnings to following vehicles, can significantly improve maintenance efficiency and potentially reduce accidents. The automotive industry is currently defining the Dedicated Short Range Communication (DSRC) standards that will enable wireless transceivers in cars to communicate with other nearby vehicles and roadside infrastructure. At the same time, the incorporation of cellular communication systems, either through dedicated in-car transceivers or by interfacing with drivers' cell phones, is also being considered. Note that while this paper uses automobile applications as a running example, similar services are also feasible with cell phones and other devices, albeit with increased energy constraints.

Since such systems try to infer information about the environment from in-car sensor originally designed for other purposes, achieving high detection accuracy can be expected to pose a challenge. Thus, we envision that these systems rely on sensing redundancy, by considering sensor readings obtained from the many mobiles that pass the same event location. This could be achieved through a naive centralized design, wherein each car communicates its readings to a server for later processing. This design, however, will make cellular channel the communication bottleneck, especially when the penetration rate, the percentage of participating cars, and the scale and sophistication of monitoring services increase. Further complicating the design is the fact that the cost of maintaining the server infrastructure (which can consist of networked servers) may also become significant. Fortunately, with increasing penetration rate, cars frequently enter each other's inter-vehicle communication range and the resulting communication capacity is comparatively large due to a high degree of spatial reuse. This observation suggests that distributed data processing and aggregation among vehicles through inter-vehicle communication can scale the system at low cost to higher penetration rates.

The resulting challenge then is the definition of communication protocols and programming abstractions that can ease the implementation of efficient data aggregation for pervasive monitoring services. This challenge fundamentally differs from conventional static sensor networking, because the specific nodes participating in the aggregation process continuously change. While sensing systems with some node mobility have already been considered (e.g., [1]), the specific challenges of highly mobile networks with nodes moving and sensing along particular routes, frequent disconnections, and continuous changes in topology remain an open problem. This challenge is also distinctly different from other mobile ad-hoc networks or peer-2-peer networks, where random node pairs communicate. While these networks also experience frequent topology changes, in our system communication is highly localized among nodes that have passed the same geographic location and thus it is beneficial to integrate location information into the networking protocols.

In this paper, we describe the Geocache programming

978-1-4244-6534-7/10/$26.00

abstraction and protocols for its implementation. Rather than storing information in a particular node, Geocache "stores" information at a particular geographic location. Geocache are implemented by letting the car that is passing a geographic location carry the information about that location. As the vehicle moves further away, it transfers the information to a following vehicle closer to the reference location. Depending on the delay constraint of the application and node density this transfer can occur at different frequencies.

The rest of the paper is organized as follows. Section II discusses the platform assumption and network architecture for pervasive monitoring applications. Section III provides a formal definition of Geocache and illustrates a few ways of using Geocache to build pervasive monitoring applications. We then describe in detail different Geocache collection protocols in Section IV. Section V compares the performance of Geocache protocols through detailed simulations and also proposes a unifying protocol that can dynamically adapt between different collection methods. Section VI discusses the related work, and Section VII provides the conclusion of this paper.

II. ROME: Assumptions and Requirements

In this section, we first discuss the state-of-the-art in-car sensing and communicating technology. Following this, we next discuss the envisioned road monitoring and alert applications that can be built upon the technology. Finally, we present the network architecture that can best support these applications.

A. Platform Assumptions

We envision that future automotive vehicles include sensing, communication and computing resources to enable sophisticated distributed sensing applications.

As far as communication is concerned, in addition to the cellular access option, the automotive industry is also defining the IEEE 802.11p standard for Wireless Access in the Vehicular Environment (WAVE) to enable vehicle-to-vehicle and vehicle-to-infrastructure communication in the 5.85-5.925 GHz band. For non-emergency applications FCC regulations permit transmission power levels of up to 2W Equivalent Isotropically Radiated Power, which would provide a free-space communication range of over 1km and about 200m (at 10% PER) in a Rayleigh fading channel. The basic CSMA/CA protocol remains virtually unchanged from 802.11a/b/g except that association and authentication procedures have been replaced.

At the same time, most vehicles also already contain a rich set of sensors that provide information about engine and emission performance as well as the environment. In the United States, the On-Board-Diagnostics-II (OBD-II) specification has been mandatory since 1996; a similar EOBD specification is mandatory since 2001 in the European Union. These specifications provide a standardized interface for access to most engine performance and emission related in-car sensors through a 16-pin J1962 connector. A sample set of accessible sensor readings are shown in Table I. The basic sensor readings shown in Table I reflect the internal settings of individual cars.

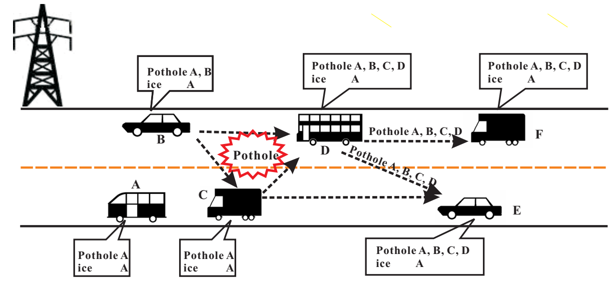


Fig. 1. Illustration of a location-centric peer-to-peer pothole detection scenario.

Additionally, modern cars also include more advanced devices which can profile the surroundings. Examples of such sensors include front- and rear-view cameras, rain sensors, wheel traction sensors, accelerometers, gyroscopes and radar/lidar systems.

The existence of sensor devices with varying capabilities can help infer the presence of high-level *events*, which may be of relevance to a large number of drivers on the road. For instance, a car can detect a pothole on the highway by a combination of observed accelerometer registered shock, sudden braking, and camera image.

B. Location-Centric Peer-to-Peer Computing in ROME

Networking cars that are equipped with on-board computing, sensing and communication capabilities can enable a family of *distributed sensing* applications, which focus on monitoring the road/traffic health by detecting abnormal events on the road. Robust event detection often requires the aggregation and mining of readings from multiple vehicles that passed the event location. For example, inferring road hazards, such as objects on the road or slippery road conditions from sudden braking on one car can lead to false alarms as such sudden braking also frequently occurs due to driver mistakes. Therefore, achieving robust detection will require the detection of clusters of braking readings around the same time and location from multiple cars.

In this study, we seek to investigate the design issues involved in developing such a **ro**ad **m**onitoring and al**e**rt system, which we refer to as *ROME*. Figure 1 illustrates a ROME scenario in which several cars collectively detect the presence of a pothole on the highway. In this simple example, cars will individually detect the pothole based on its local readings, but the base station will only confirm the

Mode (hex)	PID (hex)	Data (bytes)	Description	Min/Max	Units
01	05	1	Engine coolant temperature	-40/215	%
01	0A	1	Fuel pressure	0/765	kPa
01	0C	2	Engine RPM	0/16,383	rpm
01	0D	1	Vehicle speed	0/255	km/h
01	1F	2	Run time since engine start	0/65,535	sec

TABLE I
Five interesting sample OBD-II readings

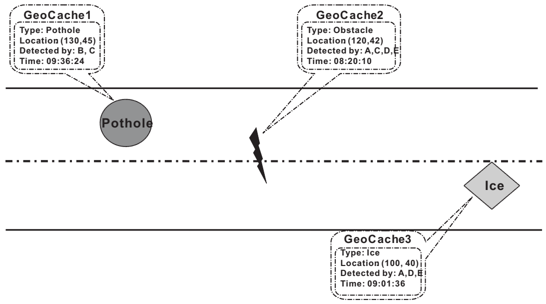


Fig. 2. Three Geocache.

presence of the pothole after gathering enough information from the passing cars. The gathering process can be as simple as counting the number of cars that detected vibrations, or as complicated as performing image recognition over several images.

Information gathering can be implemented in the following ways:

- *Centralized*: This design is based on immediately communicating all sensor readings through a cellular link to a central server, which can store and analyze the data.
- *Query-Response*: The principle of this design is to store all sensor readings locally in each vehicle, until data is requested by an outside entity. Queries must also use the cellular network, since the dispersion of cars storing similar sensor readings related to the same event increases with time and vehicular networks using short range communications with lower penetration rate remain frequently disconnected.
- *Location-centric P2P Processing*: This design involves each car that detects the event exchanging the information with its neighbors through short-ranged radio, and processing the information at the same time. Once the aggregated decision is reached, one of the cars will raise the alarm to the external server. Here, the computation is not tied to any node, but to the event location, and is carried out by the cars that pass by the location.

Among the three approaches, we advocate the location-centric P2P approach. First, compared to the other two alternatives, it provides better scalability. The centralized approach requires every car to contact the external server, which will soon saturate the cellular link given the number of events on a highway and the number of cars that pass by. The query-response approach completely ignores the importance of location in such a system; after the cars that detected the event left the event spot, it is very difficult (or, costly) to find them later to respond to the queries. In addition to scalability, the location-centric P2P approach can also help protect privacy because individual vehicle's sensor readings are not collected by a centralized entity in this approach. Recall that our example in Figure 1 employs a distributed location-centric P2P architecture.

III. Geocache: Concept and Abstractions

Before we present our location-centric peer-to-peer method of implementing ROME, we first present the concept of *Geocache*. Environmental monitoring applications such as road monitoring can be expected to exhibit strong spatial locality in data access patterns, meaning that sensor data from a particular position is frequently accessed by other vehicles passing this position. Thus, Geocache is a programming abstraction, which stores information in virtual geographic space rather than on a specific node (illustrated in Fig. 2). At a high level, the Geocache subsystem bears similarity to a distributed "Geocache database", which records observations of different spots on the highway, and can be accessed by applications through a continuous query interface. This database is unique, however, in that the actual node that stores a Geocache frequently changes as nodes pass by, to keep the information near the specified Geocache location.

To efficiently operate in a highly mobile network with frequent disconnections, (i) Geocache use opportunistic proactive dissemination methods rather than reactive query protocols; (ii) completeness of results is not guaranteed; and (iii) to reduce protocol overhead the system may transmit several Geocache in a single packet, thus all Geocache of the same type should share the same propagation parameters defined below.

A Geocache can be formally defined by the following attributes:

- *GeocacheID*,
- *Latitude*, and *longitude* which describe the geographic anchor position of the Geocache,
- *Time* which stands for the creation time of the Geocache,
- *Time-to-Live* which denotes the duration until the Geocache expires,
- *Delay* which denotes the Geocache collection interval,
- *Data* which contains an application-defined data structure.

Time-to-live and delay constitute the Geocache' propagation parameters. We note that a Geocache should have a limited lifetime to ensure limited dissemination of the Geocache. For instance, an event that was detected a long time ago should not be broadcasted among nodes. In order to ensure this temporal locality, a node invalidates Geocache if their time-to-live expired.

IV. Geocache Collection Protocols

To disseminate Geocache among the nodes, we consider *Relay Geocache* and *Boomerang Geocache* schemes and compare them to a naive broadcast baseline.

A. Broadcast Geocache Baseline

Broadcast Geocache is the baseline algorithm that can be used to store and disseminate Geocache. Every node periodically broadcasts all its known Geocache (some are generated by other nodes) until it's time-to-live expires.

Challenges: The major challenge facing this scheme is the choice of a proper broadcast time interval. Overly frequent broadcasting will cause unnecessary overhead to the system, while on the other hand, the Geocache may fail to reach other nodes if the interval is too long. Therefore, the broadcast interval should be chosen based on average node density and average node speed. Considering the node density and speed variations over different roads, however, a suitable time interval for all environments is hard to identify. Thus, in many environments periodic broadcasting will be overly aggressive, and in high density situations the system may suffer from collisions and congestion.

B. Relay Geocache

Recognizing these challenges of the naive broadcasting scheme, we propose an algorithm, where nodes explicitly pass on the Geocache to surrounding nodes just after passing the retaining location of the Geocache, which we call anchor location. This is analogues to a "relay" game, hence we refer to it as *Relay Geocache*. Compared to Broadcast Geocache, Relay Geocache is designed to incur a lower communication overhead by (i) only sending after passing the anchor location (to avoid excessive resend); (ii) using feedback from recipients to stop forwarding. We address these concepts in detail in the remainder of this subsection.

Relay Protocol. When a node receives a Geocache, it sends feedback to the previous relay node and becomes the new relay. The new relay node does not immediately forward the Geocache; instead, it carries the Geocache until it passes the anchor location, which can be evaluated by periodically comparing its current position with the anchor location. Then, the relay node constructs an new Geocache (with the same data load but re-initialized header) and sends it out to surrounding nodes. Meanwhile, it will set up a timer, and wait for feedback from the recipients. On a road with low node density, there may not be any recipients within its radio range, and the node would keep broadcasting until someone enters its radio range.

Feedback Mechanism. After the relay node sends out the Geocache and receives feedback from the following nodes, it can invalidate the Geocache from its local cache. We considered two feedback mechanisms:

- Acknowledgement. In this scheme, receivers of a relayed Geocache return an explicit acknowledgment message. When multiple nodes receive the Geocache, multiple acknowledgments are wasteful and may lead to acknowledgment collisions. Hence, we adopt a back off scheme for acknowledgments, where multiple receivers are given a different delay to send back acknowledgements. And those who overhear the first acknowledgment message would cancel their respective scheduled acknowledgment packets.
- Overhearing. An alternative to explicit acknowledgments is to overhear Geocache transmissions from successive relay nodes. Here, any node that receives a Geocache from a preceding node takes a candidate relay role, without directly responding with an acknowledgment. When it arrives at the anchor location, it will send out new Geocache, which once received by the previous relay node will serve as an implicit acknowledgment. Then, the previous relay node will overhear the new Geocache, and then delete the Geocache from its local cache.

Consider an example in Figure 3. E is the current relay node. F is behind E, outside E's communication range, and A is in

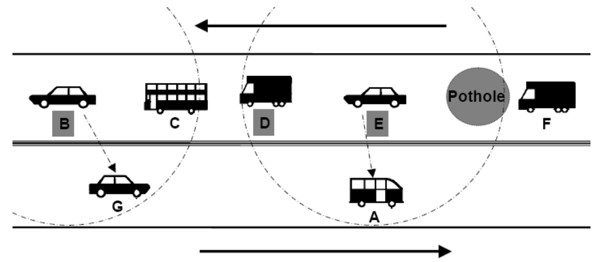


Fig. 3. Relay and Boomerang Geocache protocols.

the opposite direction, inside E's communication range. Once arriving at the anchor location, E's Geocache will be received by A, and most likely on A's way back to the anchor location, A's Geocache will be received by F. Thus after 2 rounds of hand off, the Geocache is still retained around the anchor.

C. Boomerang Geocache

The Boomerang Geocache protocol is motivated by two observations. First, many road monitoring applications can tolerate larger delays on the order of a few minutes. Second, the relay algorithm implicitly assumes that most of the cars passing the event location will detect the event. That is the reason why every car will forward the Geocache message after passing the event location. The assumption of every car detecting the event, however, does not hold in many situations—not every car may have the sensors needed to detect the event, or the highway may be much wider than the detection range of an in-car sensor (e.g., a small pothole may only affect a fraction of cars passing on one of the lanes). In these situations, involving cars that did not detect events will lead to excessive communication overhead.

To address these issues, Boomerang Geocache lets all cars passively keep their Geocache in the local cache until a collector from the opposite direction (referred to as *opposite collector*) passes to collect the Geocache. Given the Geocache delay D, the opposite collector starts collecting Geocache D time after the first car detects the event.

Consider again the example shown in Figure 3. Here, 4 cars have passed a pothole and 3 of them (B, D, E, with shadowed index number) detected its presence. After B detected the event, it refrained from transmissions and kept driving for time D, when it starts requesting an opposite collector. Since G, driving in the opposite direction, receives B's request, it becomes the opposite collector and collects the Geocache from B. G will then periodically broadcast collection requests, while driving towards the event location. Responding to these collecting request, D and E will send their Geocache to G and G will run the aggregation algorithm and dump the complete Geocache when arriving at the event location.

Boomerang Geocache intends to efficiently handle situations with low event detection rates, by avoiding frequent relaying of messages among cars that did not detect the event. The designated collector extracts information from cars, which requires only 1 hop communication with the fraction of cars that detected the event.

V. Performance Evaluation

In this section, we present our evaluation effort and simulation results.

A. Simulation Workloads and Metrics

Key factors that affect ROME's performance are node densities and relative movements particularly if the number of cars is large. To allow us to measure performance with large numbers of cars, we chose a simulation methodology based on automotive movement traces incorporated into the NS2 simulation platform. NS2 is configured to use an 802.11 MAC with two-ray-ground propagation model.

In order to reduce the simulation time, we preprocessed the trace using a perl script to extract traffic around three different locations. We first specify the location and time of first occurrence for the road event to be simulated, (x_e, y_e, t_e). Then we extract those records that satisfy the following two conditions: (1) the car generating the record will pass by the event location within a specified window after the event time, and (2) the time stamp of the record is within a specified window before and after the event time. Finally, this subset of about 1000 vehicles is fed into the NS2 simulation.

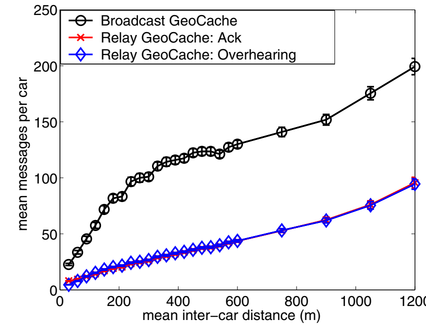
The NJTP Trace represents relatively sparse off-peak traffic from 6 to 8 AM. Since we did not have access to other data from this model, we also created a second synthetic workload to evaluate a wider range of traffic scenarios. This workload models a 70km two lane road with cars arriving in each direction according to a Poisson distribution. We vary the inter-arrival rate to model different traffic densities. In the synthetic workload, we chose 20 different Poisson arrival rates and generated 1000 scenarios for each of them (with 100 ˜ 300 cars). The average speed for each car is 30 m/s. The default radio range is 100m and will be increased to 250m for sparse traffic in Relay Geocache.

In addition to the above traffic densities, the simulation will vary the car threshold n and Geocache delay D parameters. The simulations will stop once n Geocaches are collected by one car. The baseline broadcast Geocache dissemination adopted a broadcast interval of 3 seconds (chosen as the largest interval that still ensures that all cars passing in the opposite direction can receive at least one broadcast) and a car stops broadcasting once it is 1 Km away from the event location.

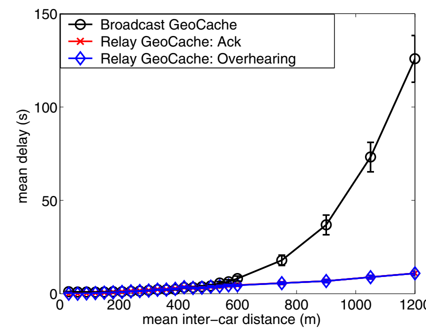
We considered the following performance metrics: (1) communication overhead—the number of packets exchanged in collecting n Geocaches, and (2) Geocache dissemination delay—the time elapsed from when the n-th car detected the event to when one car collects at least n Geocache, which is usually much smaller than D for Broadcast Geocache and Relay Geocache, but comparable to D for Boomerang Geocache.

B. Simulation Results

1) Broadcast Geocache vs. Relay Geocache.: Figures 4(a) and (b) show the results of our first set of experiments, in which we varied the car density on the road (using the synthetic workload), and confirmed an event after collecting



(a) Communication overhead



(b) Collection delay

Fig. 4. The comparison of Broadcast Geocache and two Relay Geocache schemes under varying traffic densities. The synthetic workload was used, with n=5. All the cars that pass by the event location will detect its presence.

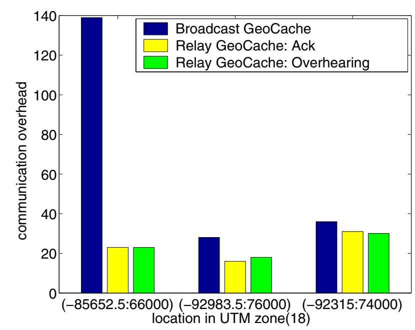


Fig. 5. The comparison of Broadcast Geocache and two Relay Geocache schemes at three locations of the real trace. We have n=5, and detection probability of 100%.

5 Geocaches. Due to the nature of broadcast, Broadcast Geocache requires a large number of packet exchanges among cars, hence has a much higher communication overhead than that of Relay Geocache. Figure 4(a) shows a 66% of improvement by Relay Geocache for an inter-car distance of 600m. On the other hand, the high communication cost paid by Broadcast Geocache does not translate into a shorter Geocache collection delay: the delays for the three Geocache strategies are comparable (Figure 4(b)).

In the second set of experiments, we compared these Geocache protocols using the real trace. We selected three random locations from the real trace, referred to as loc_1, loc_2, and loc_3, and applied the Geocache protocols to a group of cars that passed these three locations. The communication overheads of these protocols are shown in Figure 5. These results agree with the results from the synthetic trace in Figure 4(a) – loc_1

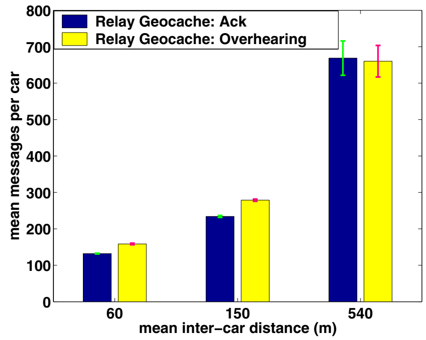


Fig. 6. The comparison of ACK-based and Overhearing-based Relay Geocache schemes . In these experiments, $n = 50$. All the cars that pass by the event location will detect its presence.

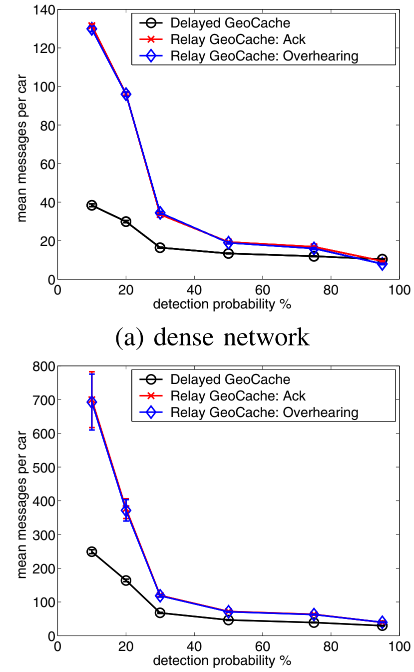
has a low traffic rate while loc_2 and loc_3 correspond to much denser situations.

These results suggest that Relay Geocache is a better strategy than Broadcast Geocache, and in the rest of this section, we will not consider Broadcast Geocache.

2) Acknowledgement vs. Overhearing.: Next, we focus on Relay Geocache schemes, and look at the difference between the ACK-based and Overhearing-based feedback mechanisms. Since the difference between these two techniques is more noticeable under a higher car threshold value (with a larger n value), in this set of experiments, we have $n = 50$. Figure 6 compares these two techniques when the average distance between adjacent cars is 60 meters, 150 meters, and 540 meters. We found that as the road becomes more sparse, the difference between these two methods first increases (the Overhearing-based technique incurs a higher communication overhead), and then decreases.

In a relatively dense network, the ACK-based Relay Geocache can provide a better feedback to the previous relay car by raising the transmission power of the ACK packets when the distance between two relay cars is large. In the Overhearing-based Relay Geocache, the previous relay car relies on the overhearing of the broadcast of the Geocache message from the following car, and it may miss the broadcast because the following car ignores the distance between cars and employs a constant transmission power. As a result, the ACK-bases scheme will perform better, and this difference becomes larger when the average distance between cars increases. On the other hand, for a very sparse network, the average distance between adjacent cars is large enough that both schemes have to rely on opposite relays to collect Geocaches. Hence, the difference between them diminishes.

3) Boomerang Geocache.: In reality, some events cannot be detected by all passing cars, and in such cases, Boomerang Geocache can be used to reduce the communication overhead. In order to validate the benefit of Boomerang Geocache, we conducted a set of experiments by varying the event detection probability, 10%, 20%, 30%, 50%, 75%, and 95%. Figures 7(a) and (b) show the number of messages exchanged per car in a dense network and in a sparse network using the synthetic trace. In both cases, Boomerang Geocache can significantly decrease the communication overhead compared to Relay Geocache schemes. For example, at the detection probability of



(a) dense network

(b) sparse network

Fig. 7. The number of total messages exchanged per car for Relay Geocache schemes and Boomerang Geocache. (a) represents a dense network in which the average distance between cars is 60 meters, and (b) represents a sparse network with the average distance between cars of 540 meters. In both cases, we have $n = 5$.

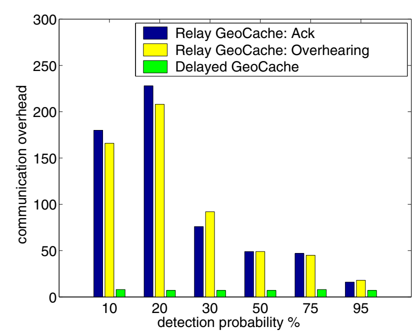


Fig. 8. The number of total messages exchanged per car for Relay Geocaches and Boomerang Geocache at location ((-92983.5, 76000)) of the real trace.

10%, Boomerang Geocache can cut down the communication overhead by 71%. Further, we observe that the communication overhead is higher in a more sparse network because the opposite collector needs to broadcast more collection request messages in these situations. The same trend is also observed in the results from the real trace (Figure 8) where Boomerang Geocache incurs a much lower communication overhead than Relay Geocache schemes.

VI. Related Work

Our work spans the fields of mobile sensor networks and vehicular networking. Perhaps the closest in spirit to the Geocache programming abstraction are geographic hash tables and spatial views. Geographic hash tables [2] provide a programming interface for data-centric storage in stationary sensor networks. In our work, we focus on programming highly mobile applications. Spatialviews [3] provides location-oriented programming language abstractions for mobile ad hoc networks, to ease application development and maintenance. This work does not address distribution of information at the protocol level, which is a key focus of this paper.

A. Mobile sensor networks.

Recent work in mobile sensor networks exploits mobility when it is not feasible to build a dense network of fixed sensors. Notably, Zebranet [4] places sensors on animals roaming in the plains of Africa, to observe their movement and socialization patterns. In under water sensor network [5], mobile nodes are robots that follow a controlled movement pattern to collect data from regions of interest. Our project differs in scale and with potentially millions of vehicle nodes participating rather than a tens or a few hundreds, and differs in their known movement patterns which are restricted by the road network. This means that typically multiple sensing units pass by and observe the same event, enabling collaborative aggregation and event detection algorithms, but also requiring efficient data distribution protocols.

Several projects target specifically vehicular sensing. CarTel [6] for example, is a comprehensive distributed mobile computing system used to collect, process and visualize data from sensors located on mobile units. It aims at exploring in-network computing on individual mobile units, as we do, but it does not use inter-vehicle communication, which in our project, is a main focus to enable distributed aggregation of sensor readings from multiple cars. Another vehicular sensor network: MobEyes [7] [8], introduces MDHP (MobEyes Diffusion/Harvesting Processor), a protocol used to spread information within WSN and build low-cost index of mobile VSN storage. Although our projects bear similarities in that we both aim to develop low-cost yet efficient inter-vehicle communication protocol, MobEyes relies largely on a opportunistically broadcast approach, possibly with the emphasis of simplified protocol, while we aim at minimizing traffic overhead by the cost of more sophisticated schemes. In VEDAS [9], the authors concentrate on mobile and distributed data stream mining system that allows real time vehicle-health monitoring and driver characterization, instead of addressing inter-vehicle communication. TrafficView [10] exploits intervehicle communication. It presents a specific application dedicated to monitoring automotive traffic congestion, while in our project, we aim to design an application-independent platform to enable collaborative sensor-driven applications on vehicles.

B. Inter-vehicle, geographic, and delay-tolerant communication.

Many projects have addressed scalable communication in mobile ad hoc networks (e.g., [11]), in sparse or disconnected mobile ad hoc networks (e.g., [1] [12] [13] [14]), or through infostations visited by mobile nodes. For example, in [15], the authors introduce **Infostations** which are lightweight infrastructures deployed in the MANET to deliver data to mobile nodes. The infostations can be envisioned as small centralized information distributers, while in our case, we don't have such centralized units and every node play the same role in propagating Geocaches. the MaxProp [12] routing protocol is used to ensure effective routing of DTN (disruption-tolerant networks) messages via intermittently connected nodes. These protocols are based on different communication workloads, such as unicast between randomly chosen nodes, or multicast to random node sets. Our protocols are optimized for communication between mobile nodes that have monitored the same environmental event. In [1], a Message Ferrying approach is proposed in which designated mobile nodes (message ferries) store and carry messages. Our project differs in that virtually all nodes are involved in communication. In [13], the authors aim to guarantee message transmission in minimal time, at the expense of additional messaging overhead. Instead, our applications are more delay tolerant, and the main goal is reducing communication overhead. In [16], the author proposes the concept of ad-hoc peer-to-peer (p2p) networking for mobile communication based on grouping. It also discusses the communication scheme carried out among vehicles traveling in opposite directions, which is also a concern in our project, but again our work is based on different application workloads.

Geocast protocols [17], [18], [19], [20] transmit messages to a predefined geographical region, which is suitable for location-based services such as position-based advertising and publish-and-subscribe. Repeated geocasts or time stable geocast [21] could also be used to maintain Geocaches in a certain area and bears similarities to our baseline scheme. It is different in concept though in that it requires the definition of a geographic region, which is not needed for Geocaches. Most geocast schemes concentrate on routing messages to the area of interest, or distributing messages to all nodes [17] and [20], while Geocaches are established close to their anchor and need only be known to at least one node. Further time-stable geocasts continuously remain in the region of interest, while Geocaches can travel away from the anchor and periodically return through the Boomerang Geocache scheme.

VII. Conclusions

In this work, we have described a distributed sensing and monitoring system for highly mobile vehicular networks. Designed for higher deployment densities, the system operates in a highly decentralized fashion, relying on inter-vehicle communication to exchange sensed and aggregated information. This obliterate the need for coordination by a centralized entity, and improves scalability by reducing load on cellular communication bottleneck.

The system builds on a novel communication abstraction that we term Geocache. This abstraction allows nodes to specify a geographic location to store information, rather than defining one particular node. The Geocache system then identifies appropriate storage nodes near the specified geographic position and hands off the information as vehicles pass by. To implement this we have proposed two protocol variants: Relayed Geocache and Boomerang Geocache, which can also be combined into a single hybrid protocol. We found that Relay Geocache reduces messaging overhead by 66%

percent and maintains low delay over a much larger range of node densities, compared to a baseline periodic broadcasting implementation. Boomerang Geocache can provide further substantial improvements, when only a small fraction of cars can detect the event. We found that it reduces messages by 71% at a detection probability of 10%.

REFERENCES

[1] Zhao, W., Ammar, M., Zegura, E.: A message ferrying approach for data delivery in sparse mobile ad hoc networks. In: MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing, ACM (2004) 187–198

[2] Ratnasamy, S., Karp, B., Yin, L., Yu, F., Estrin, D., Govindan, R., Shenker, S.: Ght: a geographic hash table for data-centric storage. In: WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, ACM (2002) 78–87

[3] Ni, Y., Kremer, U., Stere, A., Iftode, L.: Programming ad-hoc networks of mobile and resource-constrained devices. SIGPLAN Not. **40**(6) (2005) 249–260

[4] Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, L.S., Rubenstein, D.: Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. SIGARCH Comput. Archit. News **30**(5) (2002) 96–107

[5] Vasilescu, I., Kotay, K., Rus, D., Dunbabin, M., Corke, P.: Data collection, storage, and retrieval with an underwater sensor network. In: SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems, ACM (2005) 154–165

[6] Hull, B., Bychkovsky, V., Zhang, Y., Chen, K., Goraczko, M., Miu, A., Shih, E., Balakrishnan, H., Madden, S.: Cartel: a distributed mobile sensor computing system. In: SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems, ACM (2006) 125–138

[7] Lee, U., Zhou, B., Gerla, M., Magistretti, E., Bellavista, P., Corradi, A.: Mobeyes: smart mobs for urban monitoring with a vehicular sensor network. Wireless Communications, IEEE **13**(5) (2006) 52–57

[8] Lee, U., Magistretti, E., Gerla, M., Bellavista, P., Corradi, A.: Opportunistic dissemination and harvesting of urban monitoring information in vehicular sensor networks. Technical report, University of California, Los Angelos (2007)

[9] Kargupta, H., Bhargava, R., Liu, K., Powers, M., Blair, P., Bushra, S., Dull, J., Sarkar, K.: Vedas: A mobile and distributed data stream mining system for real-time vehicle monitoring. In: SIAM International Conference on Data Mining. (2004)

[10] Nadeem, T., Dashtinezhad, S., Liao, C., Iftode, L.: Trafficview: traffic data dissemination using car-to-car communication. SIGMOBILE Mob. Comput. Commun. Rev. **8**(3) (2004) 6–19

[11] Li, J., Jannotti, J., De Couto, D.S.J., Karger, D.R., Morris, R.: A scalable location service for geographic ad hoc routing. In: MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking, ACM (2000) 120–130

[12] Burgess, J., Gallagher, B., Jensen, D., Levine, B.N.: Maxprop: Routing for vehicle-based disruption-tolerant networks. In: In Proc. IEEE INFOCOM. (2006)

[13] Li, Q., Rus, D.: Sending messages to mobile users in disconnected ad-hoc wireless networks. In: MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking, ACM (2000) 44–55

[14] Briesemeister, L., Hommel, G.: Role-based multicast in highly mobile but sparsely connected ad hoc networks. In: Mobile and Ad Hoc Networking and Computing, 2000. MobiHOC. 2000 First Annual Workshop on. (2000) 45–50

[15] Frenkiel, R.H., Badrinath, B.R., Borrs, J., As, J.B., Yates, R.D.: The infostations challenge: Balancing cost and ubiquity in delivering wireless data. IEEE Personal Communications **7** (1999) 66–71

[16] Chisalita, L., Shahmehri, N.: A peer-to-peer approach to vehicular communication for the support of traffic safety applications. In: Intelligent Transportation Systems, 2002. Proceedings. The IEEE 5th International Conference on. (2002) 336–341

[17] Ko, Y.B., Vaidya, N.H.: Flooding-based geocasting protocols for mobile ad hoc networks. Mob. Netw. Appl. **7**(6) (2002) 471–480

[18] Ko, Y., Vaidya, N.H.: Geotora: A protocol for geocasting in mobile ad hoc networks. Technical report (2000)

[19] Small, T., Haas, Z.J.: The shared wireless infostation model: a new ad hoc networking paradigm (or where there is a whale, there is a way). In: MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing, ACM (2003) 233–244

[20] Morris, R., Jannotti, J., Kaashoek, F., Li, J., Decouto, D.: Carnet: a scalable ad hoc wireless network system. In: EW 9: Proceedings of the 9th workshop on ACM SIGOPS European workshop, ACM (2000) 61–65

[21] Maihöfer, C., Leinmüller, T., Schoch, E.: Abiding geocast: time–stable geocast for ad hoc networks. In: VANET '05: Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks, ACM (2005) 20–29