The Boomerang Protocol: Tieing Data to Geographic Locations in Mobile Disconnected Networks

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,tsun,gruteser,yyzhang}@winlab.rutgers.edu

***Abstract*—We present the novel boomerang protocol to efficiently retain information at a particular geographic location in a sparse network of highly mobile nodes without use of infrastructure networks. Our proof-of-concept implementation revealed the main challenge in implementing the Boomerang protocol is to accurately detect whether a node is divergent from a recorded trajectory and then followed up with a detailed study to address the challenge. Simulation with automotive traffic traces for a southern New Jersey region shows that the protocol improves packet return rate by 70% compared to a baseline implementation using shortest path geographic routing.**

I. Introduction

As daily mobile devices such as cell phones, PDAs and digital cameras start to be used as sensing devices [1], [2], [3], mobile sensing is becoming a social event instead of a high-tech phenomenon. Compared to today's special-purpose sensing applications such as automotive traffic congestion monitoring (e.g.,[4]) or pothole detection [5], mobile sensing can take place anytime, anywhere, and will have far more diverse meanings. A direct consequence of this trend is the production of a vast amount of data, in terms of both type and volume. Example data types include pictures, videos, sound files, and plain text-based sensor readings. These data can potentially bring great convenience to the society as they can serve as traces of our lives and logs of the physical world.

Due to the large volume of the data (image the number of pictures when every cell phone takes one picture per day), the up link transmission overhead could be extremely large if all data needs to be stored on remote servers. Also, most data is only useful locally, thus it is wasteful to store and retrieve data from a remote server. Finally, privacy will be a serious concern when large amounts of user data are stored in a central remote server of questionable trustworthiness.

To address this challenge, we take inspiration from real life solutions. In reality, if an item was lost/found, we post a note around the spot where it was lost/found. If we need information, we always go back to the physical location where the event occurred to search. Similarly, in the truly anytime-anywhere mobile sensing, we advocate building a "directory" [6] at each event location by having one or more mobiles near the location carry information about the data collected in the location. We refer to the directory information as *geocache*[1] of the location, and the location as *anchor*. By having the geocache always carried by the node close to the anchor, we say we can tie the data around the location where they were collected. Once the geocache for a location reaches a certain size, we can first consider compressing the information. Then we can also consider a "chaining" technique: retaining only the latest geocache entries around the anchor while leaving a link to the storage locations of older entries. Finally, we can delete old entries or unimportant ones.

In summary, the salient contributions of the work are:

- Outlining the geocache concept, making sensed data available at the anchor location, to support mobile sensing applications over a distributed network of mobile nodes.
- Designing a boomerang protocol which can periodically return geocache data to an anchor location with reduced communication overhead.
- Developing algorithms that can accurately detect whether a node is diverging from a recorded trajectory considering real-world road map complexity.
- Evaluation through a proof-of-concept implementation and showing through simulations using a portion of the south New Jersey road network that the collection and use of return trajectory information in the boomerang protocol increases the probability of timely return to the anchor location by an average of 70% compared to shortest path geographic routing.

The remainder of the paper is organized as follows. Section II discusses the related work. Section III briefly discusses the platform assumption and system model. Section IV describes in detail the boomerang protocols and the critical part of its implementation. Section V discusses our proof-of-concept implementation effort and robust techniques for detecting divergence from a recorded trajectory based on

[1] Inspired by physical geocaches that store information and items at specific locations. Finding them with GPS receivers has become a popular pastime (http://en.wikipedia.org/wiki/Geocaching).

978-0-7695-4048-1/10 $26.00 © 2010 IEEE
DOI 10.1109/MDM.2010.65

IEEE computer society

real GPS traces. Section VI compares the performance of geocache protocols through detailed simulations using realistic road maps and traffic patterns and Section VII provides the conclusion.

II. Related Work

Recent work in mobile sensor networks exploits mobility when it is not feasible to build a dense network of fixed sensors. Notably, Zebranet [1], CarTel [2], MobEyes [3], [7], TrafficView [8]. Our project differs in the way data are collected: we do not deploy nodes to collect data to a centralized server or a dedicated mobile node, but keep the data to where they are generated using nodes that are passing the location.

Many projects have addressed scalable communication in mobile ad hoc networks (e.g., [9]), in sparse or disconnected mobile ad hoc networks (e.g., [10]), or through infostations. In [11], the authors introduce Infostations to deliver data to mobile nodes. In [12], [13], the authors aim at providing location-specific information to mobile devices, in which they developed schemes for detecting and transferring information of interest. All of these techniques adopt a server-client-like approach, but in our case, the information is provided to following mobiles through other mobiles that have passed the location. The MaxProp [10] routing protocol is used to ensure effective routing of DTN (disruption-tolerant networks) messages via intermittently connected nodes.These techniques focus on delivering messages to certain nodes, while our protocols try to keep information around a certain location.

Geocast protocols [14], [15], [16] are suitable for location-based services such as position-based advertising and publish-and-subscribe. Repeated geocasts or time stable geocast [17] could also be used to maintain Geocaches in a certain area and bears similarities to our baseline scheme. Most geocast schemes concentrate on routing messages to the area of interest, or distributing messages to all nodes [14], [16], while geocaches are established close to their anchors and need only be known to at least one node.

While our geocache return protocol is inspired by delay-tolerant geographic routing [18], it is unique in recording a nodes trajectory as the node is moving away from the anchor location and using this trajectory as a guidance for selecting nodes to carry the geocache back. In [19], the authors mentioned some trajectories concepts, however, it fails to take into account the peculiarities of vehicular networks and still forwards data to a node that is physically closer to the destination only. Geopps [20] maybe is the most similar work to ours, however, it requires each mobile node to have full topology information which may not be feasible in our scenario.

III. System Model and Applications

We consider a scenario where nodes move along constrained pathways, primarily on two-way paths where nodes can move in opposite directions. Nodes can communicate via high-bandwidth short-range radios with other nearby nodes or through an intermittent low-bandwidth wide-area network. We assume that nodes have high storage capacity and are aware of their geographic position (e.g., via GPS), but that the communication system should not rely on (possibly inaccurate) road maps.

Mobile Data Management through Geocache: As mobile devices start to produce large volumes of data, efficient management of such data can bring great convenience to our everyday life. To motivate the point, let us look at the following example scenario. More application examples can be found in [21]. Alice took a picture of a car accident using her cell phone when she drove by the accident. Bob, who was involved in the accident, would like to locate such pictures and use them to support his claims in front of a police officer.

In the above example, a traditional solution likely involves Alice uploading her picture to a server and Bob downloading the picture from the server (after consulting Google). As mentioned earlier, this solution does not scale with the data volume we may expect from the truly anytime-anywhere mobile sensing. Instead, we propose a highly distributed approach in which mobile devices keep the data locally, but leave a trace around the geographic location where the data was generated.

Numerous issues need to be carefully considered in implementing this architecture. In this paper, we set out to attack the most important challenge: how can a geocache be retained around its anchor by passing mobile nodes in an efficient way?

IV. Geocache Anchoring Protocols

The goal of the geocache anchoring protocol is to retain geocache data around the corresponding anchor location while minimizing communication overhead. Intuitively, we envision the following anchoring process: the mobile node that currently carries the data (referred to as the carrier) moves away from the anchor location, either due to disconnected network or to reduce communication overhead. When possible, it will hand off the data to other nodes (referred to as the relay nodes), preferably those traveling in the opposite direction towards the anchor location. After receiving the data, a relay node will periodically examine whether another handoff is needed. This process repeats until data returns to the anchor location, and we call this protocol “boomerang” because the data returns to its origin like a boomerang.

The task of choosing appropriate relay nodes is particularly daunting because at each handoff, neither the current carrier nor the nodes within the hand off range have knowledge beyond each node’s current velocity vector. In this paper, we propose a trajectory-based selection approach and compare it to a baseline distance-based selection scheme.

Baseline Distance-Based Selection: Heuristics that fall in this category choose the node closest to the anchor location among those that are in the handoff range and moving towards the anchor location. They share the same rationale as many geo-routing algorithms such as the ones proposed in [22]. The heuristic we look at in this category is referred to as

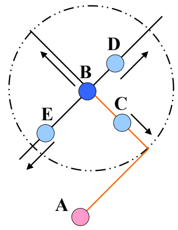

Fig. 1. An example handoff situation. In this case, B is the initial carrier. MPF will choose node E as the relay (because its distance with A, the anchor location, is decreasing and it is currently the closest to A among all the nodes), while trajectory-based schemes will choose C as the relay.

MPF (maximum progress first). A simple example is given in Figure 1.

Let us next look at the detailed handoff procedure. After traveling away from the anchor location for a certain amount of time, the initial carrier initiates a handoff by broadcasting the data along with the anchor location. Every node within the handoff range responds by checking its distance from the anchor location, and will become a candidate if its distance is decreasing. The candidate node will calculate the ACK(acknowledgement) back off time T as:

$$T = \frac{T_{max} * (d - d_0 + r)}{2r}, \quad (1)$$

where T_{max} is the maximum back off time for all nodes, say, 3 seconds; d is the distance between this receiver node and the anchor location; d_0 is the distance between the carrier and the anchor location; r is the radio radius. Using this equation, we can distribute the ACK back off times between 0 and T_{max}, and more importantly, the node with the shortest distance to the anchor location will have the smallest amount of back off time. As a result, this node will send back ACK faster, and be chosen as the next carrier.

The new relay node will keep moving until it finds out its distance to the anchor location starts increasing. At that time, it initiates another handoff procedure.

Trajectory-Based Selection: While a distance-based approach works well for geographic routing in an ad-hoc network, it may not be the most suitable one for our problem because it ignores that vehicle nodes only move along roadways. On roadways, progress in euclidian distances does not always yield a feasible path that returns to the anchor location (for instance, node E in Figure 1 may never reach anchor A).

Instead, we advocate the use of trajectory-based selection approaches. The underlying idea is the trajectory describes a feasible return path. In the same example given in Figure 1, node C will be chosen as the relay node by this scheme.

The key component of trajectory-based approaches is a "trajectory history" which stores the trajectory: the path that the data has traveled so far.

Next, let us look the detailed handoff process in a trajectory-based approach. In the discussion below, we assume the trajectory history records segments instead of continuous trace. As illustrated in Figure 2, every turn leads to a new segment which

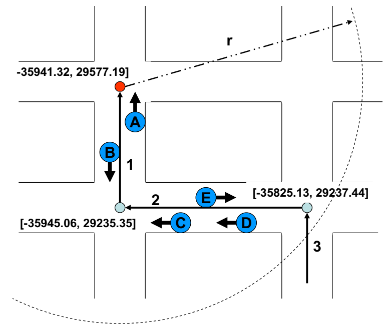

Fig. 2. Illustration of segments and trajectory-based handoff procedure.

can be represented by the coordinates of the two end points, e.g., segment 1 [-35945.06, 29235.35, -35941.32, 29577.19]. The trajectory history is implemented with a stack structure. Below is the summary of the handoff procedure:

1) *Handoff Initiation.* The current carrier broadcasts the data along with the trajectory history.
2) *Candidate Identification.* After receiving the trajectory history, every node in the handoff range pops the latest trajectory segments from the stack. We use a parameter, *lookahead distance* LAD, to control how many segments we examine. These lookahead segments can be numbered as $1, 2, ...LAD$, where segment 1 is the most recently traveled segment. In Figure 2, LAD is 3. If the node finds itself on one of the these lookahead segments, it becomes a candidate node and proceeds as below; otherwise, it just ignores the entire procedure.
3) *Candidate Prioritization.* All the candidates are prioritized according to the three rules: (1) nodes traveling in the opposite direction of the trajectory get higher priority than those traveling in the same direction as the trajectory; (2) within each direction, nodes traveling on higher segment numbers get higher priority than those on lower segment numbers; and (3) for nodes traveling on the same segment, we give higher priority to those who are closer to the anchor location.
The prioritization rules can be easily implemented if each candidate node calculates its ACK back off time using the following equation:

$$T = (\frac{LAD - s + (d - d_0 + r)/2r}{LAD} + \alpha)\frac{T_{max}}{2}, \quad (2)$$

where α is 0 if the candidate node is traveling in the opposite direction, 1 if in the same direction; s is the matching segment number; d is the distance between the candidate node and the anchor location; d_0 is the distance between the current carrier and the anchor location.
4) *Relay Selection.* The node with the smallest back off time will send an ACK first among all the candidates. To avoid the hidden and exposed terminal problem, we suggest the ACK be sent using a higher transmission

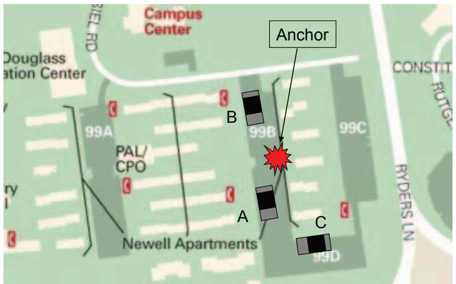

Fig. 3. Our test scenario.

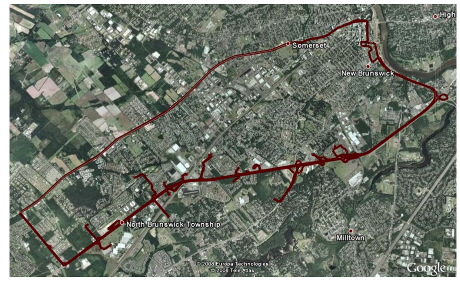

Fig. 4. The routes we traversed in the experiment are colored in red.

power so that the rest of the candidate nodes can overhear and cancel their ACKs.

5) *On Error.* If the current carrier node does not receive any ACK, it has to keep carrying the data and try to initiate a handoff periodically later.

We further distinguish between prioritized trajectory-based anchoring as described above, and a non-prioritized version, where every candidate node can choose a random back off time between 0 and T_{max}, and the winner will become the relay node.

V. Implementation and Lessons Learned

We implemented the trajectory-based Boomerang protocol with a simple testbed of three vehicles on a parking lot at Rutgers University, Cook Campus (see Figure 3).

A. Proof of Concept

Our system implementation includes several components: the sensor data collection, data storage and data communication. In the data collection component, we use a set of PERL scripts to collect data from in-car sensors and GPS device. In-car sensor information is obtained via the On-Board Diagnostic's system (OBDII) through an ElmScan 5 USB device. GPS data are collected with a Pharos USB GPS device and a Panasonic network camera WV-NM100 provides further road information that can be stored in the geocache. The data storage module uses a local SQL database (MySQL) to maintain one event table and a task table. The event table is used to store all geocache data. The task table maintains state information of the communication protocols, such as the scheduled time for handoff or the number of handoff attempts. Records of two tables are connected by the same anchor location, time of data generation, data type and host ID.

In our test case shown in Figure 3, each of the three cars carried a Linux box that is equipped with Atheros AR5212 Mini PCI 802.11a/b/g wireless cards. Two cars, A and B, each generated a piece of data when passing the anchor location. A and B passed the anchor location, and kept driving ahead with their data stored locally. Car C was driving in the opposite direction towards the anchor location. After some time, A started to hand off its data. Since C was moving towards the anchor, it caught A's data and responded with an ACK. Similarly, C also caught B's data during B's handoff. C then merged the two pieces of data, and drove with them to the anchor point.

These proof-of-concept experiments revealed that most modules in the Boomerang implementation, including trajectory recording, geocache handoff, and data aggregation worked as expected. However, they also revealed difficulties with trajectory-based decisions in the geocache return phase. Specifically, when the geocache is handed off to a return node, that node needs to keep checking whether it is driving on the trajectory but in the opposite direction. If the return node is diverging from the original trajectory, it needs to schedule another handoff. Due to inaccuracies in the recorded GPS traces and the complexity of real-world roads, the direction and divergence detection can be inaccurate. In order to improve the detection accuracy, we conducted the following additional experiments to design robust divergence detection techniques.

B. Divergence Detection Based on Real-world Traces

Divergence detection is based on a combination of distance and angle thresholds, since neither of those factors alone provides good detection across a broad range of cases: lane changes or individual's driving behavior may lead to sudden direction change without diverging, and the variance in road widths (e.g., 15–60 ft for city roads[2]) makes selection of a single distance threshold difficult. Divergence is declared if n consecutive samples meet these conditions.

$$divergence = \begin{cases} 1 & dist > dist_{max} \text{ or} \\ & dist > dist_0 \text{ and } heading > heading_0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

The key is thus to set suitable values for the two thresholds $dist_0$ and $heading_0$. In this study, we took real GPS samples and learn the appropriate threshold values based on the samples. We collected 2 hours of GPS traces in New Brunswick, NJ. Traces were collected for both highways and local streets (see Figure 4), covering about 55 miles with an average speed of 35 mph. During data collection we passed the same roads twice, once frequently turning into side streets and once staying on the main loop.

We then overlaid the two traces and manually divide them into segments so that in each segment the paths either diverge

[2]http://www.greensboro-nc.gov/visitors/

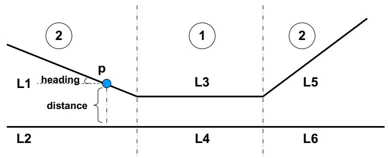

Fig. 5. Divide route pairs into 2 groups.

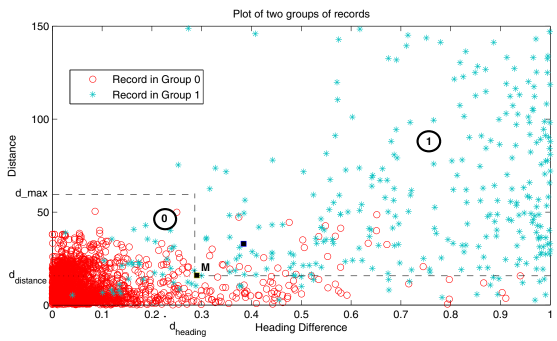

Fig. 6. The records of the two groups.

or remain parallel. This is further illustrated in Figure 5 (each segment contains one pass from each loop over the same area). The samples in each segment are then manually labeled as diverging or parallel. We then compare these labels with those generated by the detection algorithm.

Next, we divide the routes into 2 groups. Group 1 contains route pairs without divergence, and Group 2 contains route pairs with divergence. As shown in Figure 5, $(L3, L4)$ is in group 1, while $(L1, L2)$ and $(L5, L6)$ are in group 2. Take $(L1, L2)$ for example, in which $L2$ is the recorded path and $L1$ is the trace of the returning node. For each point p on $L1$, we create a new record containing four fields: $dist$, $heading$, x_1, x_2, where $dist$ is the distance from p to the original trajectory $L2$, $heading$ is the difference between p and $L2$'s direction, x_1 is the true diverges value for the current point (1 for points in group 2 and 0 for group 1), and x_2 is the predicted divergence, which is set to $Null$ at this moment. Figure 6 plots the $dist$ and $heading$ values of all the records from both groups. Group 1 and 2 are colored in blue and red, respectively. We apply clustering algorithm to both groups and have P_0 for group 1 and P_1 for group 2. Then, we take the midpoint M of the two points, and use the thresholds accordingly, $dist_0 = dist_M, heading_0 = heading_M$.

Filtering and merging the raw 1Hz samples yielded 5444 location records. With thresholds $d_{TH} = 16m$ and $a_{TH} = 0.29$, the algorithm achieves a false positive rate of 0.013 and a false negative rate of 0.187. This corresponds to an average divergence detection delay of 2.26 samples for each divergence, i.e. the divergence will be correctly detected at the 3rd sample.

VI. Simulation Evaluation

In this section, we study the performance of geocache anchoring protocols through simulations. We compare the return probability of these geocache anchoring protocols, in which the return probability is defined as the probability for a geocache successfully return to the anchor location.

A. Comparing Anchoring Protocols Using Real Road Traces

Our simulation uses realistic traffic trace collected from a south New Jersey PARAMICS model. The trace includes 984,445 records of 5000 cars in 3395 seconds during an off-peak period. We simulate communications using a free-space propagation model. In each test, we select a random car's location at a random time as the anchor location, and let that car drive for a period of T_{ho} before finding a relay node to hand off. We end the simulation when a successful return happens or until the time is over.

The return probability values for all three schemes are shown in Figure 7. Since we cannot change the node density in a real-world trace, we try to vary the radio range in this set of experiments to change the number of cars included in the hand off range. We find that except at extremely short radio radius (100m), trajectory-based schemes significantly outperform MPF on the real trace. The performance gain is around 70%. Finally, we find that all three schemes benefit from a larger radio range. This observation suggests that vehicle radios adopt a larger radius if they are to use the anchoring algorithms.

B. Combination of Distance-Based and Trajectory-Based approach

We also develop a protocol which combines these two in order to achieve the best performance. When a node receives a boomerang packet while it is on the pre-determined trajectory, it sets a flag to 1; otherwise, it sets the flag to 0. A boomerang carrier can modify the trajectory only if the flag is 1. With the above modifications, we switch from trajectory-based to distance-based. We also allow the reverse switch. This is done by comparing a certain number of trajectory segments with the carriers' current location to find out whether the carrier has returned to the trajectory. If yes, it immediately sets the writable flag to 1 and starts modifying the trajectory. In this way, it will shorten the trajectory as much as possible.

In Figure 8 and 9, we compare the performance of these three approaches. In Figure 8, we set $T_{ho} = 1000$ seconds while varying the radio range of the mobile node. It shows that across all the radio range values, the combined protocol always returns geocache with the highest probability. The trajectory based protocol has a slightly lower return probability because it misses a few opportunities to conduct a successful return if distance-based approach were used. The distance-based protocol is the worst among the three. In Figure 9, we compared the three protocols with the radio range of 750m while varying T_{ho}. We observe the same trend in this result with longer hand off delays leading to lower return probabilities.

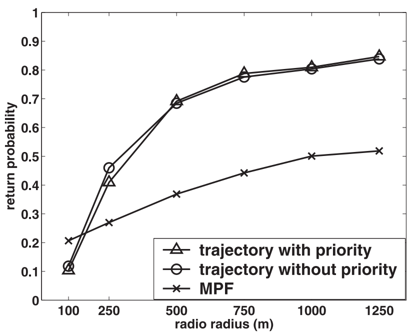

Fig. 7. The comparison of three anchoring schemes when increasing the radio range radius. $T_{ho} = 750$ seconds.

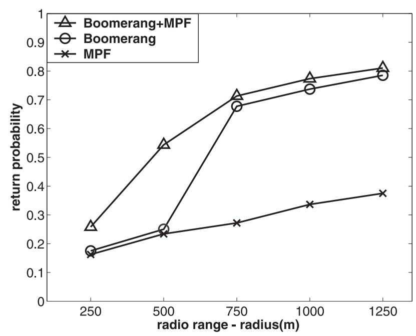

Fig. 8. Return probability when varying radio range values with T_{ho} of 1000 seconds

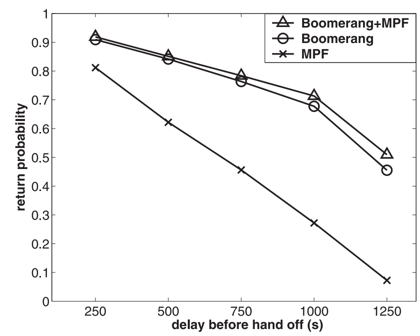

Fig. 9. Return probability when varying T_{ho} with radio range of 500 meters

VII. Conclusions

We have presented the design and implementation of the boomerang protocol to periodically make available data at a geographic location in a highly mobile vehicular network. The boomerang protocol returns data that has left its anchor location through nodes traveling in the opposite direction. To increase the probability of successful return, it records a node's path while moving away from the anchor and then uses this path to select carrying nodes to return the data. We compared this scheme with a shortest-distance georouting scheme, and demonstrate that our scheme significantly outperforms it in realistic traffic simulation, with return probability improvements up to 70%. Our prototyping revealed a number of challenges in the use of GPS trajectory data for boomerang return, which we have addressed through specific divergence detection techniques.

References

[1] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet," *ACM SIGOPS Operating Systems Review*, vol. 8, pp. 96–107, 2002.

[2] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden, "Cartel: a distributed mobile sensor computing system," in *Proc. of 4th international conference on Embedded networked sensor systems*, 2006, pp. 125–138.

[3] U. Lee, E. Magistretti, B. Zhou, M. Gerla, P. Bellavista, and A. Corradi, "Mobeyes: Smart mobs for urban monitoring with a vehicular sensor network," *Wireless Communications, IEEE*, vol. 13, pp. 52–57, 2006.

[4] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J. Herrera, A. Bayen, and Q. J. M. Annavaram, "Virtual trip lines for distributed privacy-preserving traffic monitoring," in *Proc. of 6th International conference on Mobile Systems, Applications, and Services*, 2008, pp. 15–29.

[5] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The pothole patrol: Using a mobile sensor network for road surface monitoring," in *Proc. of 6th International conference on Mobile Systems, Applications, and Services*, 2008, pp. 29–39.

[6] B. M. Oki, M. Pfluegl, A. Siegel, and D. Skeen, "The information busan architecture for extensible distributed systems," in *Proceedings of the 14th ACM Symposium on Operating Systems Principles*, 1993, pp. 58–68.

[7] U. Lee, E. Magistretti, M. Gerla, P. Bellavista, and A. Corradi, "Opportunistic dissemination and harvesting of urban monitoring information in vehicular sensor networks," UCLA, Tech. Rep., 2007.

[8] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode, "Trafficview: traffic data dissemination using car-to-car communication," *ACM Mobile Computing and Communications Review*, vol. 8, pp. 6–19, 2004.

[9] J. Li, J. Jannotti, D. Couto, D. Karger, and R. Morris, "A scalable location service for geographic ad hoc routing," in *Proc. of 6th annual international conference on Mobile computing and networking*, 2000, pp. 120–130.

[10] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, "Routing for vehicle-based disruption-tolerant networks," in *Proc. of 25th IEEE International Conference on Computer Communications*, 2006, pp. 1–11.

[11] R. H. Frenkiel, B. R. Badrinath, J. Borres, and R. D. Yates, "The infostations challenge: balancing cost and ubiquity indelivering wireless data," *Personal Communications, IEEE*, vol. 7, pp. 66–71, 2000.

[12] Y. cai and T. Xu, "Design, analysis, and implementation of a large-scale real-time location-based information sharing system," in *Proc. of 6th International conference on Mobile Systems, Applications, and Services*, 2008.

[13] H. Lu, N. Lane, S. Eisenman, and A. Campbell, "Bubble-sensing: A new paradigm for binding a sensing task to the physical world using mobile phones," in *Proc. of International Workshop on Mobile Device and Urban Sensing*, 2008.

[14] Y. B. Ko and N. H. Vaidya, "Flooding-based geocasting protocols for mobile ad hoc networks," *Mobile Networks and Applications*, vol. 7, pp. 471–480, 2002.

[15] T. Small and Z. J. Haas, "The shared wireless infostation model: a new ad hoc networking paradigm (or where there is a whale, there is a way)," in *Proc. of 4th ACM international symposium on Mobile ad hoc networking and computing*, 2003, pp. 233–244.

[16] R. Morris, J. Jannotti, F. Kaashoek, J. Li, and D. Decouto, "Carnet: A scalable ad hoc wireless network system," in *Proc. of 9th ACM SIGOPS European Workshop*, 2000, pp. 61–65.

[17] C. Maihofer, T. Leinmller, and E. Schoch, "Abiding geocast: Time-stable geocast for ad hoc networks," in *Proc. of 2nd ACM international workshop on Vehicular ad hoc networks*, 2005, pp. 20–29.

[18] E. Kuiper and S. Nadjm-Tehrani, "Geographical routing in intermittently connected ad hoc networks," in *AINAW '08: Proceedings of the 22nd International Conference on Advanced Information Networking and Applications - Workshops*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 1690–1695.

[19] J. LeBrun, C.-N. Chuah, D.Ghosal, and M.Zhang, "Knowledge-based opportunistic forwarding invehicular wireless ad hoc networks," in *Proc. of the 61st IEEE conference on Vehicular Technology*, 2005.

[20] I. Leontiadis and C. Mascolo, "Geopps: Geographical opportunistic routing for vehicular networks," in *Proc. of the IEEE Workshop on Autonomic and Opportunistic Communications(Colocated with WOW-MOM07)*, 2007.

[21] B. Zan, T. Sun, M. Gruteser, and Y. Zhang, "Room: Road monitoring and alert system through geocache," in *24rd IEEE International Symposium on Parallel and Distributed Processing*, 2010.

[22] M. Mauve and J. Widmer, "A survey on position-based routing in mobile ad hoc networks," *Network, IEEE*, vol. 15, pp. 30–39, 2001.