Extending Access Point Connectivity through Opportunistic Routing in Vehicular Networks

Ilias Leontiadis
University Of Cambridge
United Kingdom
ilias.leontiadis@cl.cam.ac.uk

Paolo Costa
Microsoft Research Cambridge
United Kingdom
costa@microsoft.com

Cecilia Mascolo
University of Cambridge
United Kingdom
cecilia.mascolo@cl.cam.ac.uk

***Abstract*—Nowadays, the navigation systems available on cars are becoming more and more sophisticated. They greatly improve the experience of drivers and passengers by enabling them to receive map and traffic updates, news feeds, advertisements, media files, etc.**

Unfortunately, the bandwidth available to each vehicle with the current technology is severely limited. There have been many reports on the inability of 3G networks to cope with large size file downloads, especially in dense and mobile settings. A possible alternative is provided by WiFi access points (APs) that are being installed in several countries along the main routes and in popular areas. Although this approach significantly increases the available bandwidth, it still does not provide a fully satisfactory solution due to the limited transmission range (usually a few hundred meters).

In this paper we present a novel routing protocol, based on opportunistic vehicle to vehicle communication, to enable efficient multi-hop routing capabilities between mobile vehicles and APs. Unlike prior work, this protocol fully supports two-way communication, i.e., the traditional vehicle-to-AP as well as the more challenging AP-to-vehicle. We leverage the information offered by the navigation system in terms of final destination and path, to i) route packets to the closest AP and ii) to route replies back to the moving vehicle efficiently.

I. Introduction

With the recent surge of fully-fledged smartphones like Apple iPhone or Android phones, people increasingly expect Internet connectivity to be available everywhere at any time. This capability is even more critical in the vehicular domain where location-based services like real-time traffic news, touristic recommendations, live map updates (possibly comprising high-resolution images as in Street View) are particularly important.

While cellular networks can be used to offer these services, this solution can also raise a number of issues. First of all, the service providers impose different rules and restrictions in each country as to what kind of data can be exchanged through their network or even what type of applications can access it, making it impossible for vehicular applications to be deployed globally. Additionally, the cost of cellular data communication is prohibitively high, as it can reach a few pence per KB. Even expensive “unlimited” plans are usually capped to a few hundred megabytes per month, making large-scale communication between vehicles unfeasible. Furthermore, even today that 3G is not widely used, the network is swamped by traffic resulting in very degraded performance in densely populated areas [1]–[3]. In addition, beside the low throughput, when moving from a cell to another, depending on the provider network policy and roaming agreements, the IP addresses are often reassigned, thus resetting all the transmissions currently in place. Given these figures, it is unclear how the 3G networks will be able to support the ever increasing data traffic without a massive investment in their network infrastructure, which is likely to further increase the cost per user.

To address these shortcomings, researchers have been looking at WiFi as a promising solution to provide Internet connectivity to vehicles. WiFi does not require any kind of licence or any infrastructure deployment. But, most importantly, local wireless communication between vehicles is becoming a reality: there is increasing interest and support for vehicular networks led mainly by the interest to maximise road safety (i.e., to avoid vehicle collisions). For instance, major car companies like Toyota and Nissan have already developed systems [4], [5] based on vehicle-to-vehicle (V2V) communication. Moreover, the 802.11 Working Group of the IEEE is developing 802.11p or *Dedicated Short Range Communications* standard (DSRC) [6] to allow V2V communication. Finally, even modern navigation systems [7], [8] include wireless interfaces (mainly to download updates when the device is taken to the owner’s house). All these are indications that in the near future short-range wireless connectivity between vehicles will be available making the deployment of *V2V applications* even more feasible.

Apart from these one-hop examples, vehicles can form larger networks where infostations and vehicles can co-operate to route and disseminate information: Infostations are fixed access points that are potentially connected to the Internet. They may act as insertion points, from where information coming from the backbone network flows towards the vehicles and vice versa. Vehicles can inter-network with each other, carrying messages even further, practically extending the range of the infostations.

In this paper, we aim to extend the coverage of the existing WiFi infrastructure through vehicle to vehicle (V2V) communication. Specifically, we focus on a query-reply interaction pattern, where the driver inputs a request (e.g., the status of the congestion on the highway or the favourite music song) and asynchronously waits for the data. This choice is motivated by the concrete difficulty of maintaining long-lived connections in such volatile environments but also by the observation that this pattern fulfils the requirements of most vehicular applications.

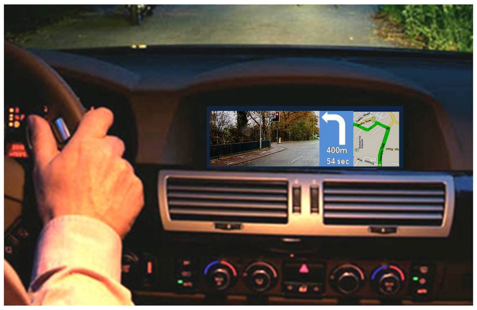


Fig. 1. Our envisioned application.

Our framework allows two operations: i) route information requests to specific geographic locations (i.e., the location of closest known infostation) in a delay-tolerant manner and ii) allow the information to be routed back to the moving vehicle. A key challenge of this approach is how to efficiently deliver the reply to the vehicle.

Indeed, previous work on routing protocols for vehicular networks [9]–[12] mostly addresses the issue of delivering the message from a mobile vehicle to a fixed destination. Here, instead, the destination itself is moving and, hence, none of these approaches would be directly applicable, and continuously tracking the vehicle's location would not be feasible either, due to the high mobility. Other solutions [13], [14] perform a restricted flood, called GeoCasting, of the reply around the last location of the vehicle hoping that one of the copies will actually reach the destination. This, however, would significantly increase the overhead, thus making this approach unfeasible for densely populated urban areas. The rationale of our approach, instead, is that the *navigation system* makes the mobility patterns predictable: the information about the car destination is usually inputted into it at the start of the journey, and the complete path is calculated and therefore known in advance.

Notice that our approach relies on routes available in the navigation systems. One might argue, however, that users tend to avoid using navigation systems for known routes (e.g., when going to their work places). Nevertheless, we expect the drivers will have an incentive to insert their destination in the navigation system as this will automatically allow them to use the extra services provided by the ad-hoc communication that are of utmost importance even for daily routes. Furthermore, systems such as Predestination [15], that can automatically detect the drivers' destination based on general driving trends and historical data for each driver, can be used to supplement or substitute the navigation system information.

II. Reference Application

In order to explain the functionality of our protocols we start by illustrating a reference application. Imagine users willing to travel through a route they are not familiar with. The navigation system already provides a valuable help by indicating what path to follow, where to turn next and on how long to stay on a specific road. However, without auxiliary information such as imaging, the task of finding the correct road to turn into or finding the desired point of interest (e.g., a shop or a restaurant) may remain challenging. This problem is even more significant if there are deviations due to temporary roadworks or accidents. Recent images of important locations and next turning points would further facilitate driving.

We envisage a system where data can explicitly be requested by drivers or implicitly by the in-car system. An example of the interface that we have in mind is illustrated in Figure 1 where the navigation system requested a fresh picture of a diversion, possibly uploaded by another vehicle in the area. Furthermore, we would like pictures of new events to be automatically routed to the vehicle.

We assume the existence of a WiFi infrastructure to which vehicles can sometimes connect to and which is connected to a central server: however, it is quite realistic to assume that no complete coverage of this infrastructure exists.

We further assume that vehicles (or, better, their navigation systems) know the location of the infostations and, at any point in time, they know which one is the closest. When in need, a vehicle may request information on a specific location or point on the route (possibly just tapping on a specific point on the screen displaying the location of interest on the map). The request is then routed opportunistically through other vehicles to the closest infostation. When there is matching content (e.g., when newer information is available) the reply is then routed back to the requesting vehicle: the routing considers the fact that this will have progressed on its path. The details of the routing protocols are distilled in Section IV-A.

Information (e.g., images) about the relevant points can be already available on the Internet, or can be uploaded dynamically on the server through an opportunistic mechanism akin to the one just described. When the data requested are not available on the server, this is sought by sending a query to the interested location: the query is routed to the location in the same way as a reply is (i.e., by reaching first the closest infostation and then opportunistically). When a vehicle in the relevant location is reached, the in-car system collects the information and sends it back to the server in the same way as an information query is sent.

III. Navigation System

Nowadays, more and more vehicles are equipped with satellite navigation systems (NS) that are typically composed of i) a GPS receiver to identify the vehicle's location, ii) maps to navigate the driver to a specific address, point of interest or location, iii) the appropriate hardware/software to aid the driver to navigate to her destination.

These systems provide turn-to-turn navigation assistance to the driver until the vehicle reaches the destination. The driver may select her destination and preferences, and the navigation system calculates a *suggested route* from the current position of the vehicle to the final destination.

To calculate the suggested route, the map of the navigation system contains statistical and historical information about speed limits, average speed, etc, and it employs a shortest-path algorithm on the road network. Furthermore, NS provides information on the Estimated Time of Arrival (ETA) or the

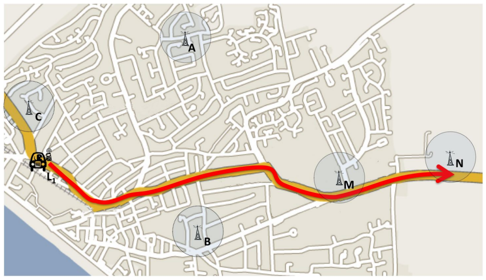


Fig. 2. Infostations are represented by antennas while the shadowed areas represents their transmissions range. L_1 is the position of the vehicle when the query is generated.

Estimated Time Required (ETR) for the vehicle until it reaches its destination.

Apart from navigation assistance to the driver, the navigation system provides valuable information such as the *suggested route*. This information *makes the mobility patterns of the vehicles more predictable* and may be used to efficiently select the most appropriate vehicles to forward and spread messages into specific geographical locations.

Secondly, the suggested routes, in conjunction with the map database (i.e., the road topology and the available points of interest), can help us to *automatically request relevant information* (e.g., information about the route/destination).

IV. Two-Way Routing Protocol

In order to correctly disseminate information, queries, and replies, our framework needs to incorporate a routing protocol capable of i) routing queries towards a static destination (e.g., the closest infostation or a vehicle in a given area) and ii) routing replies back to the *moving* vehicle. For the former task, in principle, any position-based routing protocols could be used. We chose GeoOpps [10], a protocol we developed, which enables efficient geographic forwarding in vehicular networks by exploiting information available in the navigation system.

Conversely, the latter task, i.e., propagate the reply back to the querying vehicle, resulted more challenging because the geographical position of the destination changes along time and, hence, none of the available position-based solutions, including GeoOpps, would suit.

Therefore, we designed and implemented a novel routing protocol: The key intuition at its basis is that the expected route of the requesting vehicle can be extracted by the navigation systems and piggybacked in the query message. In this way, once a reply is generated, it can be forwarded towards the *route* so as to intersect the vehicle's expected path.

In the rest of this section, we outline the main characteristics of our protocol, encompassing both information/query and reply routing.

A. Algorithm Overview

1) Query: When a a vehicle V wants to receive any kind of information from the backbone, it needs to issue a request. The generated query Q is injected in the system that isolates the closest infostation and issues a request message which will be routed to its geographic location. An example is provided in Figure 2 where infostation C is selected as the closest to the current position of the vehicle. The request message also *contains V's suggested route to destination*. This information can be extracted from the navigation system. Afterwards Q is opportunistically routed (either by hopping from vehicle to vehicle, or by being carried by the same car) to the selected infostation.

2) Reply: When there is matching content or the reply is ready we need to route it back to the vehicle. However, since the vehicle is constantly moving, we need to make sure that the information is sent on the forecasted route of the vehicle.

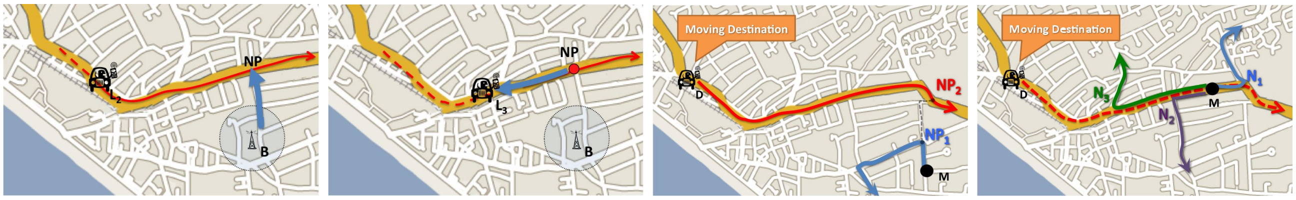
1) When the reply R, containing the data requested, is ready, our system selects an appropriate infostation to inject the answer into the vehicular network. This infostation will be the one that is estimated to be the closest and beyond the requesting vehicle V considering its route. Note that the infostation may not be on the requesting vehicle's path, as we will use the vehicle-to-vehicle communication to fill this gap. For example, in Figure 2 infostation B will be selected as it is close to V's path and ahead of its route.
2) The reply R is then opportunistically routed (V2V) from the infostation to a point *on vehicle V's route*. Our aim is to deliver R on the path ahead of the vehicle and keep it there in order to maximise the chances of delivery. Furthermore, the reply travels towards the vehicle (always on the path) so as to minimise delay.
   In Figure 3 we show an example on how the reply is routed back to the vehicle from an infostation which is not on the vehicle's path. Initially (Figure 3(a)) the information is routed from the selected infostation B to intercept the route of the vehicle at point NP.
   When the packet arrives to NP it then starts to move towards the vehicle but always on the route. This is shown in Figure 3(b) where the packet is routed from NP backwards along the route until it meets the vehicle at location L_3.
3) Eventually, the vehicle is reached and the data delivered (or it expires).

V. Protocol Description

We now describe the protocol steps in detail.

A. STEP 1: Query Routing

GeoOpps [10] is a delay tolerant routing algorithm that exploits the availability of information from the navigation system in order to opportunistically route a message to a certain geographical location. Messages are hopped from vehicle to vehicle in an ad hoc manner. GeoOpps takes advantage of the navigation system's suggested routes to select the carrier vehicles that are likely to take the information closer to the final destination of the message, each time. GeoOpps operates through the following steps:



(a) Intercepting route.

(b) Keeping the message on route and moving backwards.

(c) Neighbour selection when outside route.

(d) Neighbour selection when on route.

Fig. 3. Routing the reply. L_2 and L_3 indicate the vehicle's progressive positions. NP is the nearest point between the vehicle's route and the infostation.

- Firstly, for a candidate vehicle V, it finds the nearest point NP on the vehicle's navigation route that has the smallest *driving distance* to the message's destination D.
- Afterwards, it calculates an *utility function* U_v that expresses the Minimum Estimated Time of Delivery: $U_v =$ *Driv. Time to NP* $+ \alpha \times$ *Driv. Time from NP to D*. This utility estimates how quickly the message can be delivered if V becomes the next carrier. Obviously this depends on its future navigation route (taken by its navigation system).
- It forwards the message if the neighbour has *smaller utility*, i.e., if the packet could be delivery sooner because the neighbour is going closer to D.

The protocol was conceived with the aim of allowing opportunistic message delivery in a decentralised network of vehicles. In this paper we exploit it to deliver the request to the closest infostation (e.g., infostation C in Figure 2) from which the query will then be handled as a regular IP packet.

B. STEP 2: Reply Infostation Selection

When the reply is available we need to find the infostation from which to inject it into the vehicular network. As we described before, the information about the route and position of the vehicle when the request was sent, which is piggybacked on the request packet, and the information about predicted travelling times on roads, are used to determine the best infostation to receive the reply data. The selected infostation is the closest and *beyond* the vehicle's estimated position. This selection allows to route so as to intercept the vehicle's path before it arrives.

More precisely, to evaluate whether an infostation can potentially deliver the message to the vehicle:

- Find the Nearest Point (NP) on the route of the vehicle to the infostation (see example in Figure 3(a));
- Calculate an estimation of the time required t to route the message from the infostation to NP (using the GeOpps utility function);
- Calculate the time t_v when the vehicle will arrive to NP based on the information from its navigation system (i.e., the time required for the vehicle to drive from L_2 to NP in the example);
- This infostation is a candidate if the time t taken by the message to reach the vehicle's route is shorter than the time the vehicle would take to reach this point (t_v). If multiple infostations fulfil these constraints, the one with the lowest t_v is chosen in order to minimise the time for the vehicle to reach NP.

Once the infostation is chosen, the reply message is routed there through the backbone network. An example is provided in Figure 3: infostation B is the closest infostation to the vehicle route among those depicted in Figure 2 and, hence, the reply originates there.

C. STEP 3: Reply Opportunistic Routing

Once the data message is on the closest infostation, it will be routed to the vehicle's path. There are two sub-steps here: 1) the message has to be routed on the path (Figure 3(a)) and then 2) the message has to be *kept on the route* and preferably move backwards to meet the vehicle (Figure 3(b)).

1) Intercepting destination's path: This time we need to route the message towards a whole path [1]. In this case, we need to find carriers that can intersect the destination's navigation path (between the current estimated position and its final destination).

As we observe in Figure 3(c), there can be up to *two nearest points*: NP_1 on the neighbour's path and NP_2 on destinations' path [2]. These two points are selected to minimise the driving time between *the two paths* (dotted line in Figure 3(c)). Then our framework estimates the driving time t_1 required from the current position to NP_1 and, afterwards, the time t_2 required from NP_1 to NP_2. Finally, only if the message is estimated to be at NP_2 earlier than the vehicle-destination then it is considered for forwarding.

2) Keeping the message on path: The second sub-step starts once the reply has reached the route, it is presumably beyond the vehicle's position (e.g., Figure 3(b)). Our target at this stage is primarily to *keep the message on the route* to maximise delivery ratio. Nevertheless, to minimise delay, our framework hops the reply backwards on the requesting vehicle's route until the vehicle is reached. When a vehicle carrying a reply meets a neighbour it evaluates whether or not to forward a message based on how much the route of the neighbour overlaps with the destination route:

- It checks if the neighbour is on the destination's path.
- It finds the last way-point P before it will deviate from the destination's vehicle path. For example, in Figure 3(d) the green vehicle (marked as N_3) will stay on D's path until location N_3.

[1] GeOpps is only used to deliver a packet to a specific geographic location.
[2] These two points can overlap when the two routes cross each other.

- It estimates the time t_{vn} that destination D will take to reach P (e.g., driving time between D and N_3 in the previous example).
- It calculates its own last point O before deviating and the time t_v it will take for the requesting vehicle to reach O;
- If t_{vn} is lower than t_v, then the neighbour is a better carrier and the message is handed over. Intuitively, this means that the neighbour will carry the message closer to D before deviating from D's route. Otherwise the message stays on the current vehicle.

An example is sketched in Figure 3(d). Once message M has reached the vehicle's route (black dot in the picture), three different neighbours, respectively N_1, N_2, N_3, are available. Among these, N_3 is selected because, as illustrated in Figure 3(d), it is going in the right direction (i.e., towards the vehicle) and its expected route has a higher overlap with the vehicle's one. Along the path, other neighbours may appear and the message will be transferred to one of these if it is going closer to the destination until the message is finally delivered. Sometimes it may occur that a vehicle holding the message is deviating from the destination's navigation path and there is no suitable candidate. In this case, our protocol re-initiates the procedure to bring the message back to the route as described above in the case of the infostation.

VI. Related Work

A large body of work appeared in literature addressing geographic routing in ad hoc networks [14], [16] and, lately, in vehicular networks (e.g., [9]–[12]). Unfortunately, none of these approaches fully satisfy our requirements as all of them assume that the geographic position of the final destination, be an area or a specific host, is known a priori. Vice versa, in our scenario, replies must be forwarded to a mobile vehicle and, hence, its position is continuously changing. In addition, while most approaches adopt a push-based interaction style, we opted for a pull-driven paradigm that seems to suit better the applications we have identified.

In the close research area of Delay Tolerant Networks (DTNs) [17], several routing protocols have been devised to deliver messages in a store-and-forward fashion based on opportunistic contacts [18], [19]. These protocols exploit different mechanisms to route a message to the destination such as statistics of previous encounters, or precise mobility schedules to find the best routes. However, none of them has been specifically designed for vehicular networks and it is not clear how they can cope with the peculiar mobility patterns of these networks.

Finally, some recent papers (e.g., Drive Through Internet [20] and Cabernet [21]) focus on techniques to improve the 1-hop communication between vehicles and infostations, by operating several optimisations at both the link layer and transport layer. While the goals of such work differ from ours, these techniques are complimentary to our approach and integrating them in our framework is part of our future research agenda.

VII. Conclusions

In this paper we have introduced a two-way routing protocol for hybrid vehicular network that enables extending the access point connectivity through opportunistic routing. We showed how we can exploit the navigation system to predict mobility and route messages. As future work we are investigating a possible multi-request, multi-reply model which allows aggregation of replies and content caching. Finally, we would like to fully evaluate our approach both by implementing and testing our system in real settings and by running simulations.

Acknowledgments: We acknowledge the support of EPSRC through Project CREAM.

References

[1] "AT&T Chief: Operators aren't Prepared for Onslaught of Data Traffic," May, 2009, http://tinyurl.com/at-t-chief.

[2] "Can O2 Cope With Smartphone Traffic?" 2009, http://tinyurl.com/n56esl.

[3] "iPhone 3G network issues frustrating early adopters," 2008, http://tinyurl.com/5akkf4.

[4] C. L. Robinson, L. Caminiti, D. Caveney, and K. Laberteaux, "Efficient Coordination and Transmission of Data for Cooperative Vehicular Safety Applications," in *VANET '06: Proceedings of the 3rd international Workshop on Vehicular Ad Hoc Networks*. New York, NY, USA: ACM, 2006, pp. 10–19.

[5] Nissan Motors Co., Ltd., "Nissan to Test Intelligent Transportation System In Kanagawa," 2009.

[6] IEEE P802.11-Task Group P, "Wireless Access for the Vehicular Environment (WAVE)," 2009.

[7] TomTom Navigation Systems, http://www.tomtom.com.

[8] Dash Navigation, Inc., http://www.dash.net.

[9] J. Lebrun, C.-N. Chuah, D. Ghosal, and M. Zhang, "Knowledge-Based Opportunistic Forwarding in Vehicular Wireless Ad Hoc Networks," in *Proc. of VTC-Spring*, 2005, pp. 2289–2293.

[10] I. Leontiadis and C. Mascolo, "GeOpps: Opportunistic Geographical Routing for Vehicular Networks," in *Proc. of the IEEE Workshop on Autonomic and Opportunistic Communications*, 2007, pp. 1–6.

[11] H. Fubler, H. Hartenstein, D. Vollmer, M. Mauve, and M. Kasemann, "Location-Based Routing for Vehicular Ad-Hoc Networks," in *Proc. of MobiCom*, 2003, pp. 47–49.

[12] C. Lochert, H. Hartenstein, J. Tian, H. Fuler, D. Hermann, and M. Mauve, "A Routing Strategy for Vehicular Ad Hoc Networks in City Environments," in *Proceedings of the IEEE Intelligent Vehicles Symposium 2003*, Columbus, OH, USA, June 2003, pp. 156–161.

[13] Y.-B. Ko and N. H. Vaidya, "Flooding-Based Geocasting Protocols for Mobile Ad Hoc Networks." *MONET*, vol. 7, no. 6, pp. 471–480, 2002.

[14] C. Maihöfer, "A Survey of Geocast Routing Protocols," *IEEE Communications Surveys & Tutorials*, vol. 6, pp. 32–42, 2004.

[15] J. Krumm and E. Horvitz, "Predestination: Where Do You Want to Go Today?" *Computer*, vol. 40, no. 4, pp. 105–107, 2007.

[16] M. Mauve, J. Widmer, and H. Hartenstein, "A Survey on Position-based Routing in Mobile Ad-Hoc Networks," *IEEE Network*, vol. 15, no. 6, pp. 30–39, 2001.

[17] S. Jain, K. Fall, and R. Patra, "Routing in a Delay Tolerant Network," in *SIGCOMM '04: Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. New York, NY, USA: ACM, 2004, pp. 145–158.

[18] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: Modeling a Three-tier Architecture for Sparse Sensor Networks," in *Proc. of Sensor Network Protocols and Applications*, 2003, pp. 30–41.

[19] W. Zhao, M. Ammar, and E. Zegura, "A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks," in *MobiHoc '04: 5th int. Symp. on Mobile Ad Hoc Networking*. New York, NY, USA: ACM, 2004, pp. 187–198.

[20] J. Ott and D. Kutscher, "A Disconnection-Tolerant Transport for Drive-Thru Internet Environments," *INFOCOM 2005*, vol. 3, pp. 1849–1862 vol. 3, 2005.

[21] J. Eriksson, H. Balakrishnan, and S. Madden, "Cabernet: Vehicular Content Delivery Using WiFi," in *MobiCom '08: 14th ACM Int. Conference on Mobile Computing and Networking*. New York, NY, USA: ACM, 2008, pp. 199–210.