Evaluating the feasibility of using smartphones for ITS safety applications

Sergio M. Tornell, Carlos T. Calafate, Juan-Carlos Cano and Pietro Manzoni
Universitat Politècnica de València
Camino de Vera, s/n, 46022 Valencia, Spain
sermarto@upv.es, {calafate, jucano, pmanzoni}@disca.upv.es
Manuel Fogue, Francisco J. Martinez
University of Zaragoza, Spain
{m.fogue, f.martinez}@unizar.es

***Abstract*—Driving security and comfort can be improved by applying Intelligent Transportation Systems (ITS) proposals. The low adoption rate of new ITS hardware and software products is slowing down the market introduction of these solutions. In this paper we present a driving safety application for smartphones based on a warning dissemination protocol called eMDR.**

The use of smartphones minimizes the hardware cost and eliminates most of the adoption barriers; users will no longer have to install new dedicated devices in their vehicles. Instead, they will simply have to install an application in their smartphone. Our application is integrated with a Navigation System which provides access to road maps, current location, and route information.

We analyzed the behavior of the wireless channel and the GPS location service under different conditions to assess the feasibility of our proposal. Results showed that, in C2C communications, smartphones are able to provide a reasonable degree of connectivity, and that the degree of precision achieved is enough for certain types of driving safety applications.

***Index Terms*—smartphones, VANET, broadcast, warning message dissemination, real maps**

I. Introduction

Wireless networks have evolved at an incredibly fast rate, being applicable to several contexts and different communication solutions. Vehicular Ad-Hoc Networks (VANETs) allow vehicles to communicate with other vehicles in their neighborhood and with the infrastructure. For the automobile industry, many on board security systems and wireless solutions have been proposed to improve safety-related and data communications. Concerning safety-related communications, vehicles have different sensors which collect information not only about engine status, or speed, but also context information (e.g. weather or traffic status). In this situation, vehicles can benefit from a collaborative distribution approach, harvesting this information through the Vehicular Ad-Hoc Network. This collaboration may result profitable when a local event, such as an accident, the detection of an abnormal situation on the road, or an incoming ambulance, occurs and must be notified to nearby vehicles. The information treatment and management is closely related to the idea of SmartCities [1] [2], a set of initiatives proposed by the European Commission to increase the efficiency of our cities through the use of technology.

According to the European Automobile Manufacturers Association [3], the average age of the car fleet in Europe is 8.7 years, and 34.5% of the automobile fleet in the EU are older than 10 years. Taking these statistics into account, if manufacturers started to install specialized On Board Units (OBUs) in every car right now, in the best case, it would take more than 10 years to achieve a penetration rate of about 80%. In addition, experience demonstrates that only luxury cars tend to incorporate these hi-tech devices as standard equipment. In our opinion, traffic management and safety-related applications cannot wait until then. Simultaneously, smartphones recently reached a 50% of penetration in developed countries, and this value is still growing. Smartphones are typically equipped with several network interfaces: WiFi, cellular network, and Bluetooth. From our point of view, smartphones offer an opportunity for developers and VANETs, which can evolve from a pure ad-hoc network, with its known limitations, to a heterogeneous and more versatile network taking advantage of the possibilities offered by other wireless network interfaces. However, the use of smartphones also presents some drawbacks related with interferences due mainly to the use of the ISM band as it is stated in [4].

In this article we describe a solution that exploits the direct communication between smartphones through an ad-hoc network to provide a better driving experience when integrated together with a navigation software. As a first approach, we have built an application that implements our protocol "enhanced Message Dissemination based on Roadmaps" (eMDR) [5] to inform its users about the presence of an ambulance close to their route; this allows the drivers to act in consequence, anticipating their actions to clear the way for the ambulance.

The rest of the paper is organized as follows: section II reviews applications that attempt to improve the user experience through collaboration. We then describe our proposed architecture in section III. In section IV we present our experiments, which evidence the feasibility of our solution. Finally, we remark our conclusions and discuss future work in section V.

II. Related Work

In the last few years, VANETs have received much attention from academia, industry, and governments. Due to the high cost of deployment, VANET protocols and applications are usually tested through simulation. In [6] authors presented an example of a broadcasting protocol for VANETs that allows to disseminate information efficiently. Another example of a

simulation based study is [7], where the authors presented a Car-to-Car (C2C) communication protocol. On the other hand, there have also been several projects that developed their own testbed platform based on dedicated hardware; CarTel [8] uses nodes deployed in vehicles and sporadic connections to open access points for the purpose of monitoring and classifying road surface conditions. However, vehicles do not communicate between them. In [9] researchers from UCLA presented CVet, a VANET testbed deployed over vehicles belonging to the UCLA car fleet. As far as we know, these and other solutions presented for fast prototyping and testing in VANETs are built over dedicated hardware, which increases the cost of deployment and impede their adoption.

Regarding navigation devices, we have found different solutions that aim at improving the user experience while driving and moving efficiently. Recently, navigation devices manufacturers, like TomTom [10] or Garmin [11], have presented their own *real time assisted routing* based on online updates provided via a cellular network. According to information made available at their websites, they combine data obtained from the local authorities with data obtained from each individual device. Their devices do not have any wireless interface for ad-hoc communication, so they miss the opportunity of real collaboration with neighboring vehicles. Moreover, the updates via cellular network may present a high delay.

Concerning mobile applications, in [12] its authors demonstrated how sensors available in smartphones can be used to prevent accidents in a proactive manner, through the analysis of the data collected by cameras, accelerometers and the Global Positioning System (GPS) system. Furthermore, there are several Android based applications [13] that try to make driving a better experience. The best example is Waze [14], an application that exploits the user collaboration through the cellular network. Another example of the possibilities that smartphones and their multi-interface schemes can offer to VANETs is Torque [15], an application that can connect to the OBD-II [16] interface of the vehicle to obtain real-time information about the current state of the engine.

Up to now, none of the currently available solutions takes advantage of ad-hoc communication techniques. To the best of our knowledge, this is the first proposal adopting smartphones as a suitable platform for the quick development and prototyping of VANET applications.

III. Architecture Overview

We have built our VANET application system upon standard smartphones based on Android. With this election we avoid the requirement of a special approval for installation in vehicles, and we also take advantage of the relatively low price of these devices. Moreover, the number of smartphones already sold reduces the cost of deployment, since we do not need to install new hardware on every vehicle.

Applications will directly use the Android socket API for network communications, and the file input/output API for the logging system.

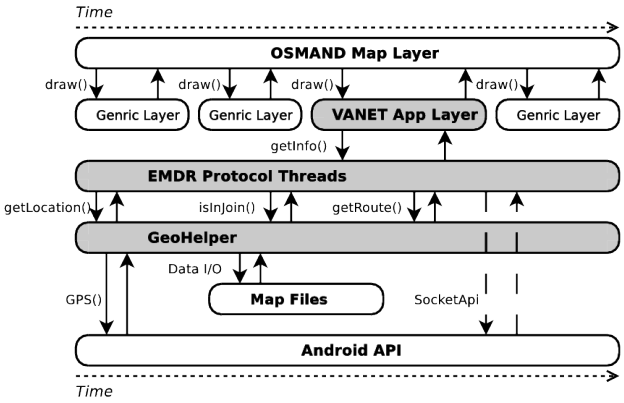


Fig. 1: Architecture overview.

A. Wireless Radio

Smartphones are usually equipped with an 802.11b/g interface that, in our case, has been configured in ad-hoc mode. The typical transmission power of these devices is 16 dBm (40mW), which is far from the maximum allowed transmission power for IEEE 802.11g in Europe (20 dBm, i.e. 100 mW). These values are really low compared with the maximum transmission power for IEEE 802.11p (33 dBm, i.e. 2 W), which is the standard PHY/MAC protocol for VANETs. It is worth noticing that the use of the ad-hoc mode avoids any setting up delay due to the association and authentication processes.

B. Navigation Software

To take advantage of available software, we decided to integrate our VANET application architecture with an existing open source navigation software. Our main requisites were: (i) to have a free map data source to avoid royalties issues, (ii) to have an offline route calculation system, and (iii) to present an easily expandable structure. With these premises in mind, after scouting the Android market, we chose OsmAnd [17], a navigation software that uses maps and route layout information from OpenStreetMaps [18]. The map rendering process in OsmAnd is composed by different layers that are rendered sequentially. Therefore the different applications can be programmed as a special layer that not only draws new data on the map, but also communicates with the dedicated protocol threads which use the socket API to communicate with other vehicles. We have developed a class, called 'GeoHelper", to simplify the use of geographical data provided by the integrated GPS, and to deal with map issues related with the navigation service. Figure 1 shows how our application is integrated into the OsmAnd architecture. Summarizing, the OsmAnd map layer calls the *draw()* of our new VANET App Layer, which obtains the required info from the different protocol threads, eMDR in this case. On the other side, the protocol threads communicates with our GeoHelper class, which runs under its own thread, to obtain geographical info.

IV. eMDR overview

eMDR uses location and street map information to facilitate an efficient dissemination of warning messages in VANETs,

avoiding the well known broadcast storm problem and taking into account the effects of buildings to avoid wasting transmissions. In this section we include a short description of the behavior of eMDR, a more detailed description of eMDR can be found in [5].

In eMDR, vehicles operate in either *warning* or *normal* mode. The normal mode represents a default behavior; however, when a vehicle detects a dangerous condition, it will start operating in the warning mode. Warning mode vehicles inform other vehicles about abnormal situations by sending warning messages periodically (every T_w seconds). We consider abnormal situations as any condition that could affect the traffic security and probably cause an accident, e.g., slippery road, a previous accident where the involved vehicles are an obstacle for the normal traffic flow, works on the road, etc. Normal mode vehicles enable the diffusion of these warning packets and, periodically (every T_b seconds), they also send $beacons$ with non-critical information such as their positions and speed. Normal messages are not propagated by other vehicles. With respect to warning messages, each vehicle only propagates them once for each sequence number, i.e., older messages are dropped.

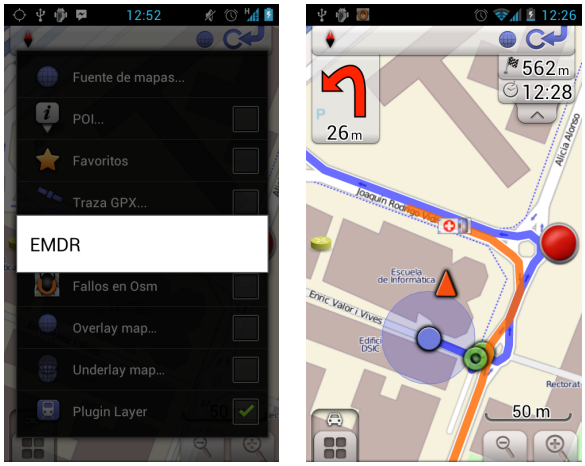
In order to disseminate warnings as quickly as possible, eMDR works as follows: When $vehicle_i$ starts the broadcast of a message, it sends m to all of its neighbors. When any nearby vehicle receives m for the first time, it rebroadcasts it by further relaying m to its neighbors. Depending on their characteristics, every vehicle repeats the $send(warning)$ or the $send(beacon)$ operations periodically with different periods (T_w and T_b, respectively). When a new message m is received, the vehicle tests whether m has already been received. To evaluate this condition, each vehicle maintains a list of message IDs. An incoming warning message ID is inserted in the list if m is received for the first time (i.e., its ID has not been previously stored in the list), and it is rebroadcasted to the surrounding vehicles only when the distance d between sender and receiver is higher than a distance threshold D, or the receiver is in a different street than the sender. We consider that two vehicles are in a different street when: (i) both are indeed in different roads (this information is obtained by on-board GPS systems with integrated street maps), (ii) the receiver, in spite of being in the same street, is near to an intersection, or (iii) the receiver detects that it has neighbors in different streets. Hence, warnings can be rebroadcasted to vehicles which are traveling on other streets, overcoming the radio signal interference due to the presence of buildings. If the message is a $beacon$, it is simply discarded since we are not interested in the dissemination of beacons.

V. Implementation Details

The Intelligent Transport Systems (ITS) application that we have implemented broadcasts the location and the route of an approaching ambulance using the enhanced Message Dissemination based on Roadmaps (eMDR) protocol warning mode, informing nearby drivers, which will be able to act accordingly to favor the progress of the ambulance toward its destination. In this section we briefly describe the implementation details of our application user interface, our eMDR protocol and our *GeoHelper* class.

A. User Interface

As explained above, our application is implemented as a new layer that adds information obtained from the VANET to the map view in OsmAnd. Before implementing our application we have also created a new interface called *GeoPluginLayer* that offers a common structure to child classes. This allows us to quickly implement different protocols reusing common code. Figure 2a shows the interface through which users will select the desired protocol, in our case eMDR. Figure 2b shows our "Warning Ambulance Application" running. We can see two idle neighbors represented by green circles, and a neighbor in alarm mode, represented by an ambulance icon; the orange line is the ambulance's route, and the blue line represents the vehicle programmed route. Both the red button on the right and the route-shaped button on the left, are used for testing purposes. The red button activates the alarm mode and, if present, broadcasts the programmed route; it is supposed to be available only to authorized devices, such as ambulances, police-cars, etc. The other one is used to select between three forwarding modes: (i) normal forwarding, *i.e.* following eMDR rules, (ii) unconditional forwarding, *i.e.* every alarm message is rebroadcasted, and (iii) forwarding disabled, *i.e.* no alarm message is rebroadcasted.



(a) Layer selection screen. (b) Application screenshot

Fig. 2: Different application screenshots.

B. eMDR Implementation

Following a divide and conquer paradigm we have structured our implementation of eMDR in three different classes: an upper class, called EmdrPluginLayer, that handles *onDraw()* calls from the map and all user interface related events (buttons). It makes use of two privates threads that implement the eMDR protocol, namely: EmdrPacketGenerator and EmdrProtocol. The former is in charge of packet generation, while the latter is in charge of receiving, processing, and forwarding the messages received from other nodes following eMDR rules. Our eMDR protocol will send a beacon every

second, or warning messages instead when in alarm mode. In addition, if a route is programmed, it will be added to the warnings as an array of points.

C. GeoHelper Class

To simplify the management of geographical data we have developed a class called "GeoHelper". This class collects and processes data from files and from the services provided by both OsmAnd and the Android operating system (*i.e.* GPS, routing service, etc), offering different methods to our application.

The most important method provided by our GeoHelper class is the method *getCurrentLocation()*, which returns, an estimated current location based on the last two updates provided by the GPS interface and the direction of the current road. The difference between the last two updates is used to estimate the speed vector of the vehicle, if a *getLocation()* call occurs between two consecutive updates; the current position is estimated using the estimated speed vector and the last known location. In the case of *getLocationOnStreet()* calls, the estimated location is restricted to be on the closest road, and the speed vector also lays on the current road.

Concerning the *findRoute(location)*, we found that the time required by OsmAnd to calculate a route between two locations was in the order of tens of seconds, and that this value is strongly dependent on the number of possible routes. In our opinion, these issues impede the usage of some routing protocols that calculate the shortest route for every sent packet, as in [19].

VI. Evaluation

The application described in the section V has been tested in a real scenario. Due to the existing limitations when evaluating real implementations (measuring small delays is almost impossible due to clock drift effects, making it impossible to obtain the exact position of a mobile node for a given time T), we decided to focus on the effects of the wireless channel and the behavior of the GPS interface.

A. Devices' detailed description

For our experiment we have selected three different devices from the same manufacturer: HTC. In an attempt to prove that very advanced smartphones are in fact not required for these applications, we chose three devices whose performance varies a lot, and that are definitely not the most recent smartphones currently in the market:

1) HTC Desire: Released in 2010, 1Ghz CPU core, 512 MB Ram, WIFI driver: bcm4329.
2) HTC Hero: Released in 2009, 588 Mhz CPU core, 288 MB Ram, WIFI driver: tiwlan1251.
3) HTC Tatoo: Released in 2009, 528 Mhz CPU core, 256 MB Ram, WIFI driver: tiwlan1251.

B. Experiments

We have designed a set of experiments to evaluate the following performance parameters: (i) message reception probability when in Line Of Sight (LOS), (ii) message reception probability when nodes are in different streets, and (iii) GPS updates inter-arrival time. All the experiments were performed in a real environment: vehicles were parked in streets with a typical traffic flow. Since all mobiles were inside vehicles, the transmissions were also affected by different issues related to adverse signal effects caused by the structure of the vehicle.

1) Message reception probability when in LOS: In this experiment we placed two of the handsets in different cars; then, using our application, sent a burst of 200 warning messages and counted the number of messages successfully received. We have executed two different experiments to evaluate this metric. The first one was executed in a high traffic density environment, and the second one was in a low traffic density environment. To achieve comparable results each experiment was repeated four times, therefore we have represented the mean and the confidence interval at 95%. Results, represented in figure 3, shows that, as expected, the reception probability decreases when the distance increases. Comparing both graphs, we can appreciate that the presence or the absence of interferences due to traffic density can highly influence the performance of VANET's application in smartphones, reducing the communication rate from 80 m to merely 40 m, and increasing the variability of the results. Our experiments have shown that the eMDR threshold distance (*i.e. the minimum distance at which a retransmission is worth*), is optimal when the reception probability at such distance is of 40%. Therefore, we have chosen 60 m as the threshold distance for our eMDR protocol.

2) Message reception probability when in N-LOS: In this experiment the cars where located in perpendicular streets. One of them was located 25 meters away from the intersection, and the second vehicle was moving away from the intersection. Figure 4 represents the location of the cars. As expected, the moving car stopped receiving messages as soon as it moved a few meters away from the intersection. With these results in mind, we decided that the threshold distance under eMDR to consider that a vehicle is "near to an intersection" would be configured to 10 meters or less. Also, experiments have shown that the best parameter to detect if a vehicle is close to an intersection is the detection of neighbors from different

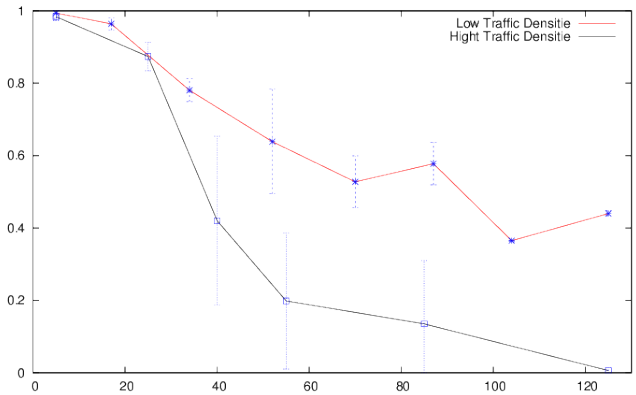


Fig. 3: Rx Probability vs Distance

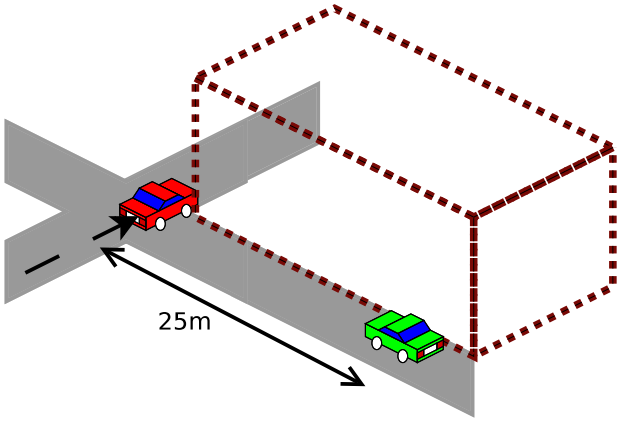


Fig. 4: Location of the cars for NLOS measuring.

streets. This method avoids problems related to the street layout representation format of OsmAnd.

3) GPS updates inter-arrival time: Another important issue when checking the feasibility of our solution is the freshness of the GPS data in smartphones. In the rest of our experiments we collected around 12000 measurements for this metric. By analyzing them, we found that the mean inter-arrival time for GPS updates was 1.07 s, while the maximum value was of 15 s, and only in 1 % of the total test its value was different from 1.0s. Although we have configured the GPS interface to notify our application about location changes as soon as possible, the minimum time between updates that the system was able to provide was of 1.0 s. If we consider that, a vehicle with a speed of 25 m/s, a maximum acceleration of 0.8 m/s^2, and a maximum deceleration of 4.5 m/s^2 can typically travel between 21.40 m and 25.40 m per second, our position estimation system, which assumes a constant direction and speed during the inter-update time, will introduce a maximum error of 3.6 m due to mobility. We believe that this value is small enough to be used in VANETs.

VII. Conclusions and Future Work

In this paper we showed that the use of smartphones opens a new horizon for VANET applications. We presented an implementation of an ITS warning service based on VANETs for Android smartphones. The selected application advertises ambulance's warnings to nearby vehicles, and it can be easily extended to support any type of warning alert such as a slippery road, or an accident. This application offers a smart interface to geographic information through the integration in a Navigation Software, thus taking advantage of more than the mere geographic position. We have demonstrated through measurement that, in C2C communication, smartphones provide a reasonable connectivity and enough precision for information dissemination, although the use of the 2.4 GHz band makes them not suitable for highly delay constrained applications. The designed architecture also allows the quick development and testing of VANETs applications.

In the near future we will explore the possibilities that emerge when evolving from a pure ad-hoc network to an heterogeneous and more versatile network, especially by taking advantage of other network interfaces. Moreover, we will explore the fusion of other sources of data, like the integration with the OBD-II system of the vehicles.

Although security issues are outside of the scope of this paper, spoofing attacks, where a non-authorized entity sends warning packets, could be avoided by using an asymmetric keys encryption scheme to sign every warning packet sent by authorized entities. We will carefully address this problem and its different solutions in future work.

Acknowledgments

This work was partially supported by the *Ministerio de Ciencia e Innovación*, Spain, under Grant TIN2011-27543-C03-01.

References

[1] "European commission's website for smartcities," http://eu-smartcities.eu/mobility_transport, last visit Jul 2012.
[2] "Esafety website," http://ec.europa.eu/information_society/activities/esafety/index_en.htm, last visit Jul 2012.
[3] "European automobile manufacturers association," http://www.acea.be/news/news_detail/vehicles_in_use/, last visit Jul 2012.
[4] W. Vandenberghe, I. Moerman, and P. Demeester, "On the feasibility of utilizing smartphones for vehicular ad hoc networking," in *ITS Telecommunications (ITST), 2011 11th International Conference on*, aug. 2011, pp. 246 –251.
[5] F. J. Martinez, M. Fogue, M. Coll, J.-C. Cano, C. T. Calafate, and P. Manzoni, "Evaluating the impact of a novel warning message dissemination scheme for vanets using real city maps," in *Proceedings of the 9th IFIP TC 6 international conference on Networking*, ser. NETWORKING'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 265–276.
[6] L.-C. Tung and M. Gerla, "An efficient road-based directional broadcast protocol for urban vanets," in *Vehicular Networking Conference (VNC), 2010 IEEE*, dec. 2010, pp. 9 –16.
[7] R. Frank, E. Giordano, P. Cataldi, and M. Gerla, "TrafRoute: A different approach to routing in vehicular networks," in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2010 IEEE 6th International Conference on*. IEEE, Oct. 2010, pp. 521–528.
[8] J. Eriksson, H. Balakrishnan, and S. Madden, "Cabernet: vehicular content delivery using wifi," in *Proceedings of the 14th ACM international conference on Mobile computing and networking*, ser. MobiCom '08. New York, NY, USA: ACM, 2008, pp. 199–210. [Online]. Available: http://doi.acm.org/10.1145/1409944.1409968
[9] M. Gerla, J.-T. Weng, E. Giordano, and G. Pau, "Vehicular testbeds; validating models and protocols before large scale deployment," in *Computing, Networking and Communications (ICNC), 2012 International Conference on*, 30 2012-feb. 2 2012, pp. 665 –669.
[10] "Tomtom," http://www.tomtom.com/, last visit Jul 2012.
[11] "Garmin," http://www.garmin.com/, last visit Jul 2012.
[12] A. Campbell and T. Choudhury, "From smart to cognitive phones," *Pervasive Computing, IEEE*, vol. 11, no. 3, pp. 7 –11, march 2012.
[13] "Google play," https://play.google.com/store, last visit Jul 2012.
[14] "Waze," http://www.waze.com/, last visit Jul 2012.
[15] "Torque," http://torque-bhp.com/, last visit Jul 2012.
[16] International Organization for Standardization, "ISO 15765: Road vehicles, Diagnostics on Controller Area Networks (CAN)," 2004.
[17] "OsmAnd website," http://www.osmand.net, last visit Jul 2012.
[18] "OpenStreetMap website," http://www.openstreetmap.org, October 2012.
[19] X. Wang and C. Song, "Distributed Real-Time Data Traffic Statistics Assisted Routing Protocol for Vehicular Networks," in *Parallel and Distributed Systems (ICPADS), 2010 IEEE 16th International Conference on*. IEEE, Dec. 2010, pp. 863–867.