

Document downloaded from:

<http://hdl.handle.net/10251/78946>

This paper must be cited as:

Fernandez Laguia, C.J.; Cano Escribá, J.C.; Tavares De Araujo Cesariny Calafate, C.M.; Manzoni, P. (2016). Experimental Evaluation of a Low-Cost Digital Sign-Posts Architecture for ITS Applications. En Ad-hoc, Mobile, and Wireless Networks. Springer. 294-307. doi:10.1007/978-3-319-40509-4_21.



The final publication is available at

https://link.springer.com/chapter/10.1007/978-3-319-40509-4_21

Copyright Springer

Additional Information

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-40509-4_21

Experimental Evaluation of a Low-Cost Digital Sign-Posts Architecture for ITS Applications

Carlos Fernandez-Laguia, Juan-Carlos Cano, Carlos T. Calafate, and
Pietro Manzoni

Universitat Politècnica de València,
Camino de Vera, s/n, Valencia. SPAIN
<http://www.grc.upv.es/>

Abstract. Integrating road signs information is becoming a critical goal for Intelligent Transportation Systems (ITS). Unlike other driving automation features, this capacity requires not only the vehicle, but also posts and infrastructure to be adapted thus involving an investment that can only be justified by a substantial number of users.

In this paper we describe an architecture that aims to facilitate the introduction and deployment of this technology based on low cost devices as the digital sign-posts and the integration of smartphones as an alternative in-vehicle user-interface. Wireless communications based on IEEE 802.11 is used for the basic connectivity requirements.

From the results obtained through an experimental evaluation, we show that, despite the smartphone constraints, we can achieve successful detection and recognition experiences at up to 90 km/h. Ultimately the experiment described confirms that the use of smartphones represents an opportunity to expand wireless technology in the traffic sign digitalisation area.

Keywords: Digital traffic signs, Smartphones, Intelligent Transportation Systems, Vehicle-to-infrastructure communications

1 Introduction

Automotive industry has made considerable advances over the last years in the fields of safety and comfort. Some of the main contributors have been the features categorized as Advanced Driver Assistance Systems (ADAS), a large number of which are already commercially available. Despite ADAS has been proved as a good opportunity to decrease the number of road accidents [1], usage rate is still significantly low. One of the reasons is that although automotive development cycles have been reduced in the last years, and in-cycle actions represent now a major update on the product concept, the vehicle replacement time is still lengthy. In the European Union the average for a passenger vehicle replacement is approximately 9.5 years while in the United States the figure exceeds 11 years. On the other hand recent surveys pointed out that a lack of perceived usefulness is one of the main reasons why awareness of ADAS is still significantly higher

than its usage rate [2]. Therefore new vehicle features do require a considerable amount of time to become a technology of common use.

This is especially critical when the technology update does not only involve the vehicles, but the roadway infrastructure. This is the case for applications like CoMoSeF [3] or traffic management systems [4] that are design to exchange information with roadside stations. In this category of communication systems, which is known as Vehicle-to-Infrastructure (V2I), only a significant amount of users can justify the investment to adapt the road according to the requirements. In particular using wireless technologies to exchange data with road infrastructure requires each traffic sign (or group of them under certain conditions, like its relative position) to be equipped with a network adapter.

In this paper we describe an architecture that aims to facilitate the introduction and deployment of this type of service which is based on low cost devices as the digital sign-posts and the integration of smartphones as an alternative in-vehicle user-interface. Wireless communications based on IEEE 802.11 is used for the basic connectivity requirements. To the best of our knowledge, there is no previous empirical experiences introducing the smartphone as a key element in the V2I strategy.

The paper is organized as follows. Section 2 describes the relevant related work on the topic. Section 3 details the proposal and Section 4 presents the experimental set-up details. Finally, Section 5 describes the evaluation results and Section 6 the final conclusions.

2 Related Work

New automobile safety features like Intelligent Speed Adaptation (ISA) or Traffic Sign Recognition (TSR) are increasingly highlighting the importance of bringing traffic signing into the digital domain in an efficient manner. This trend will only be reinforced in a future where transportation is outlined by more connected and more autonomous vehicles.

The problem has been broadly addressed in the literature using different approaches. Among them, the most extended proposal exploit enhanced vision techniques that have been adapted to road environments. Another system that has made its way to real applications is based on digital maps that contain relevant data and location references, so the client can query speed limits and other traffic information based on a known position. Although these solutions have reached a certain degree of popularity, there are some intrinsic limitations that make them not suitable for a driverless scenario. For instance light conditions and partial occlusions are two important handicaps for camera-based systems. Further, in the case of digital maps, an accurate location awareness and database update frequency are two major concerns to be addressed.

The solution that has been recently identified as a better candidate applies wireless technologies to exchange road information with vehicles. This approach appears for first time in the literature on [5]. Yoshimi Sato and Koji Makanae took the opportunity that camera-based systems were not yet extended in Japan

and proposed a different solution: equipping the road with general-purpose RFID tags that contains relevant traffic sign information. The main issue observed by other researches about Satos' work is the position of the RFID tags and the antenna. The communication range of the type of RFID devices used by Sato is approximately 40 cm.

Later proposals achieved better results and versatility, for example on [6], the authors increased the communication range up to approximately 30 meters, allowing an easier installation of the communication devices on both sides, infrastructure and vehicles. However the approach of wireless technologies had not been extensively studied until the vehicle-to-vehicle communications (V2V) concept appeared in the literature. Along with the vehicles, traffic signs have also been considered to be equipped with a wireless interface [7]. A new 802.11 amendment was published in 2010 motivated by a quicker data exchange between mobile nodes within a maximum range of one kilometer [9].

Despite the progress in this area, when we compared the cited proposals for wireless data exchange with other techniques they all share two common disadvantages: (1) the requirement of dedicated equipment and (2) the infrastructure investment. Vision systems have already established a strong position in the automotive market, therefore the adoption of a more expensive technology is not an easy challenge. Moreover this cost is not only applied to the vehicle price, but it also requires a major change in the infrastructure, which can only be justified by an important number of users. In this work we proposed an alternate platform to the in-build capture device that aims closing the gap between technical prerequisites and the requirement of potential users to easy an infrastructure change.

3 Proposed Architecture

Following our main objective, which is providing an efficient solution at a low cost, we have based our proposal on extended wireless communication standards rather than specialised protocols. The idea behind is creating a basic model that could be implemented on different platforms. We set our target on fulfilling major requirements to transmit road signs information via wireless media and support future updates. Consequently we have chosen to draft a proposal that allows high compatibility at different stack layers.

The communication model is the classical client-server scheme where road signs act as service providers, offering traffic data to the various clients concurrently upon request. The communications topology has been conceived as a traditional infrastructure network under a transport layer where the messages are transmitted via TCP. Finally, a REST-like protocol has been drafted to exchange information at the application layer.

The Medium Access Control (MAC) services are exploited at the client application level, managing the connection among different traffic signs. The mobile device inside the vehicle listens to the wireless medium for beacons. Each traffic sign, which has been previously equipped with the proposed server solution, pe-

riodically sends beacons with the relevant information for the connection. When the vehicle approaches a traffic sign, the device receives this beacon and immediately sends an authentication/association request (see Fig. 1). After the connection is established, the client can access the relevant data from the traffic sign and its location. Because the mobile device continuously monitors the signal strength received from the traffic sign, these values can be used as a reference to determine whether the vehicle is approaching or leaving a particular area. The reliability of this strategy still needs to be studied under certain conditions like shielding or multi-path; however based on our experience, we believe that this data can be exploited for that purpose on its own or to complement other real-time location systems.

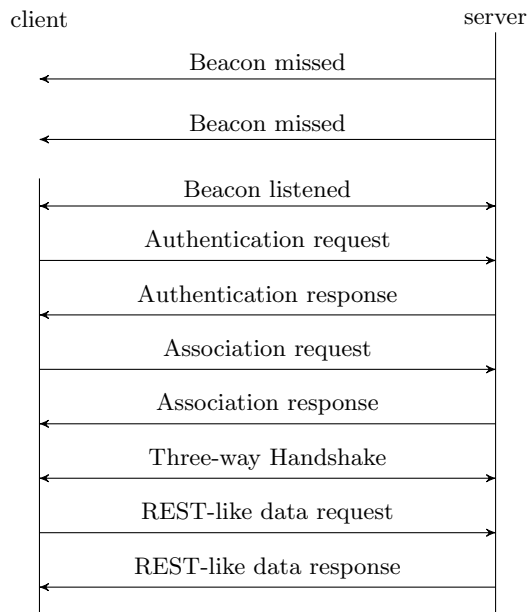


Fig. 1. Proposed communication flow diagram

It should be underlined that the proposed architecture is intended to be implemented with only two requirements: (1) a platform able to exchange data and manage the connection with different nodes in a specific order and (2) a network adapter that handles the connection at a lower level. This is specially important at the client side, which aims at gaining availability by integrating the complete solution on a smartphone.

On the server side, this architecture propitiates the convenient conditions to reduce the requirements for computing power, and therefore to lower the battery

consumption. Consequently there could be a significant cost reduction if we use a low-cost solution to equip traffic signs with a wireless interface.

4 Experimental Set-up

The idea that we propose could potentially pull ahead the benefits of wireless communications applied to traffic sign recognition by offering the driver a interim solution to adopt this technology regardless of the vehicle age. In order to demonstrate that the proposal accomplishes the basic requirements we have built an experimental solution, which has been used for the field application test that is explained on Section V.

In order to keep costs low we chose to base our experimental solution on well-proven open source software. Fig. 2 shows a simplified representation of the different software packages used on the field application test. Both sides of the systems have been implemented on Linux-kernel-based operating systems. In particular, the server side uses a *Debian* distribution where we have installed a *hostapd* daemon that handles the 802.11 standard services to connect with the different clients. The network layer is managed by the *dnsmasq* application, which is used as a light DHCP server. Finally high layer messages are managed by our own script, which implements the REST-like protocol, and incorporates a Python library for the TCP/IP standard. Among the hardware solutions in the market that fulfil our requirements, we have chosen the *Raspberry Pi*. Its wide range of connectors, specially general-purpose inputs/outputs (GPIO), along with the fact that it uses a Linux-kernel-based operating system makes it a versatile and robust solution for our objective. The network adapter is marketed under the name RTL8188CUS-GR, an USB device included in most of the *Raspberry Pi* development kits, which provides an inexpensive and complete solution for the 2.4GHz band. For the trials we configured the network manager to work on 802.11n mode.

The client side implementation relies on the *Android* API (Application Programming Interface) to communicate with the *wpa_supplicant* at the user space. The supplicant handles the connection with the different traffic signs including applying communication filters based on the beacon information. This is significantly important because the supplicant should dismiss any traffic signs that does not contain relevant information for the driver. We have considered this topic as a subject of future work, hence it is not expound it further in this text.

The model we presented is not restricted to be implemented on a smartphone and even less on a specific operating system. We have considered that the smartphone has become a commodity of widespread adoption among drivers, hence a solution based on smartphones could significantly increase the number of users, if does not cause any negative effect on the device performance. Additionally as an electronic platform, it offers flexibility and portability at a relatively low cost. Among the mobile device operating systems we chose *Android* because it led the market share of 2015. Its availability and familiarity among the potential users makes it a convenient platform to speed up the adoption of this technol-

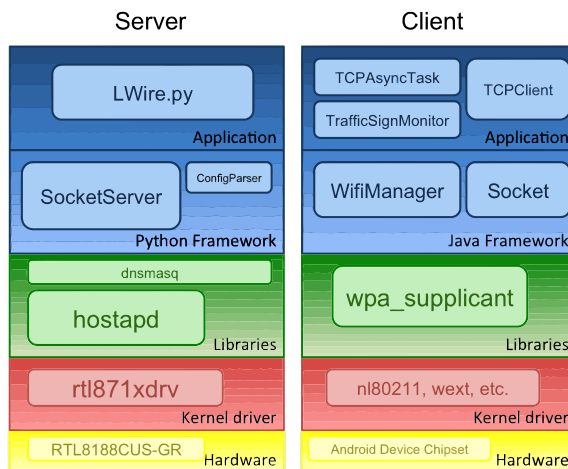


Fig. 2. Simplified network stack representation based on open source software

ogy. It should be noted that despite its benefits we are sometimes limited by the *Android* interface as it is described in the next section.

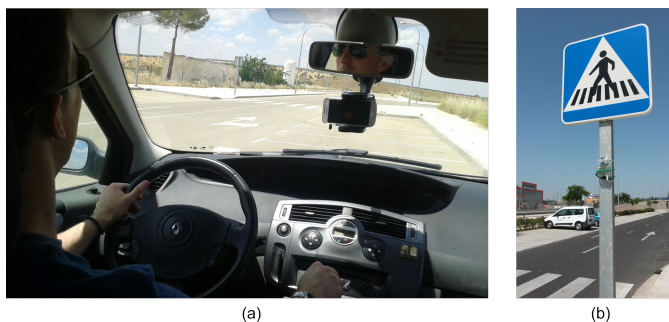


Fig. 3. (a) Smartphone attached to the vehicle windshield during the experiment. (b) *Raspberry pi* solution installed on a zebra crossing sign.

For further analysis we have developed an application that does not only retrieve information from the traffic signs in the surroundings, but also implements a minimal user interface, monitors the smartphone performance and measures the process cycle time. *Android* has been designed as an operating system for touch screen devices, hence the importance of the user interfaces. This might be a problem executing time-consuming network commands. To overcome this situation we have implemented the main logic of the application on a separate thread designed as a finite-state-machine as we can see on Figure 4.

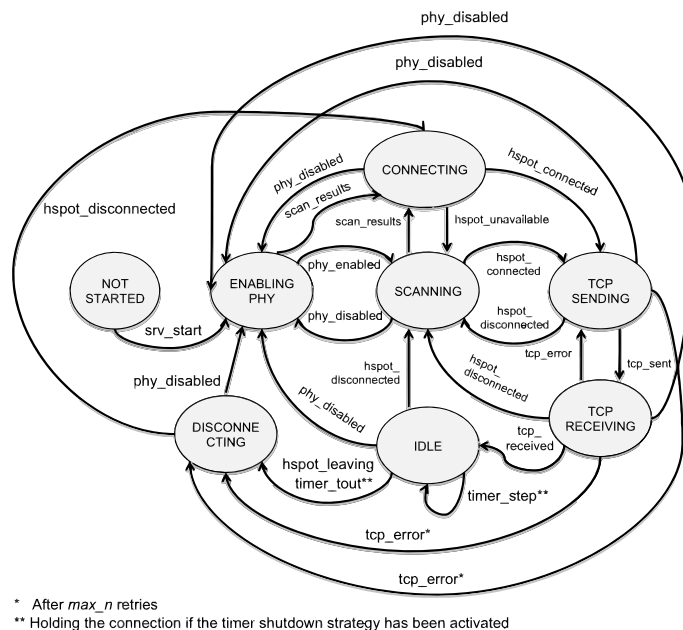


Fig. 4. Our proposal application state machine diagram

We installed the application on a *GT-I9195*, also known as *Samsung Galaxy S4 mini*. This smartphone with a dual core at 1.73 GHz and 1331 megabytes of RAM belongs to the *Samsung* lower cost lineup that reached a certain degree of popularity in 2013 because of its price. The device is shipped with a Li-Ion battery, which offers autonomy of 1900 milliamp-hour. The operating system running on the device is *Android 4.4.2*, labeled as *KitKat*. This version of the system is the last one that *Google* released with the virtual machine *Dalvik*, however it is still the most extended one among the *Android* users (87.6% according to official sources).

For our experiment we installed the smartphone on a car mount attached to the vehicle windshield where driver can see it without affecting his/her driving ability as shown in Fig. 3. Also we attached a *Raspberry Pi*, which had been previously set up with a wireless network adapter and an external battery, to a traffic sign at approximately 1.5 meters from the ground. Along with our application we additionally downloaded and installed other three traffic *Android*-based solutions that helped us to create a more complete overview on the efficiency of our application by doing separate trials on each one.

The applications installed were: *myDriveAssist* from *Robert Bosch GmbH*, which uses the built-in camera from the smartphone to detect traffic signs; *aCo-Driver* from *Evotegra*, an independent company that combines camera and GPS (Global Positioning System) data to retrieve the speed limits from the road; and finally *Michelin Navigation* a navigation system, which implements a mode



Fig. 5. Satellite image from area where experiments have been performed. (source: *Google Earth*)

where only speed limits and not directions are prompted to the driver relying on a remote database.

5 Results

The evaluation for our proposal has been carried out by two different experiments. The first one aims at proving that our solution is able to detect, register and prompt relevant data to the driver before the vehicle overtakes the traffic sign. In the second one we wanted to establish whether the proposal was efficient from the perspective of a resource-constrained device.

For the first experiment we assumed the maximum communication range between traffic sign and client invariable. The distance travelled by the smartphone when the information is prompted to the driver would then be determined by the vehicle speed and the required time for a complete data exchange between client and server. The distance is given as:

$$d = R - \frac{s}{t_{\text{detection}}} \quad (1)$$

Where R is the maximum communication range and s the vehicle speed. On the one hand the speed entirely depends on the driver and the traffic conditions; therefore we have considered different speed values for our experiment. On the other hand we wanted to establish the magnitude of the effect that this speed has on the data exchange time; with this purpose the client application logs time stamp and location at the different stages of the process (vehicle moving, traffic sign detected, etc.). The results from the first experiment are shown in Fig. 6

From them we can conclude that higher speed values do not necessarily increase the time that the smartphone needs to complete traffic sign detection, so for simplicity we can assume that its influence is limited to the amount of time available for detection. However the detection time shows a wide variability regardless the vehicle speed. In order to understand the requirement on cycle time

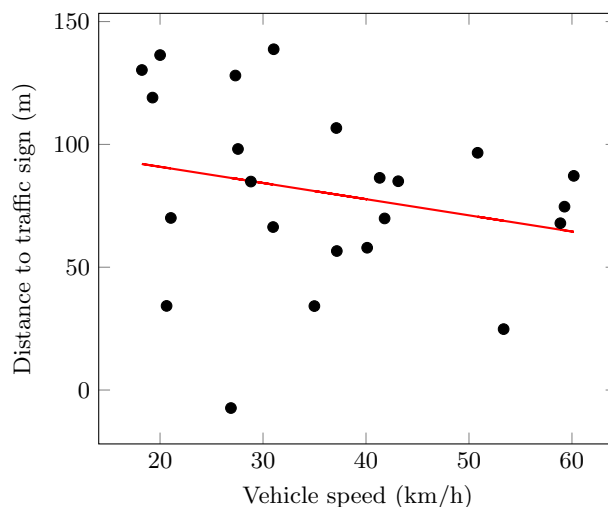


Fig. 6. Relation between vehicle speed and traffic-sign distance.

for the data exchange, we modified the source code on the *Android* application to create a log file with the timestamps at critical process steps like request for traffic sign discovery, authentication, connection, etc. We installed the modified application on the *GT-I9195* and run a trial on the simulated environment in the lab. Both devices, smartphone and digital post were placed approximately one meter from each other.

The results in table 1 shows that after approximately one hour execution 74 detections have been successfully achieved and more than two thirds of the average time per cycle was spent on the discovery phase. This phase gets initiated when the application calls the `startScan()` method and ends either when the network driver performs a full scan or a known profile is available for connection.

Table 1. Detection time break down

Execution phase	Average time (ms)
scan	2859
authentication	134
association	830
Rest-like	483
disassociation	18

From the *wpa_supplicant* documentation¹ we have learnt that every time the application calls the `startScan()` method the network driver is requested to

¹ https://w1.fi/cgit/hostap/plain/wpa_supplicant/

perform a full scan, which in the 2.4 GHz band consists of 13 channels spaced 5 MHz apart. The driver is usually designed to stay on each channel a certain amount of time before switching to the next one. The reason is that access points send periodically a beacon with relevant information for the connection, without that beacon the communication cannot be established. The full-scan latency is therefore determined by:

$$scan_latency = N \times beacon_interval \quad (2)$$

This beacon interval is typically set up to 100 milliseconds on the access point (AP) [10], however most of stations (STAs) use a more conservative value to ensure that the beacon does not get missed if the antenna is physically located within the communication range. For example on *WEXT*, one of most extended kernel drivers for Linux, this value is defined as 250 in the source code by the constant `WEXT_CSCAN_PASV_DWELL_TIME`, therefore a full scan could take up to 3250 milliseconds.

The *wpa_supplicant* and most popular drivers for *Android* devices allow minimising the scan delay by reducing the number of channels, however the *Android* API accepts no parameter for the `startScan()` method. This is, indeed, an important inefficiency compared with other solutions. Dedicated applications, like the ones based on RFID technologies, do not really require a full scan since transmitter and receiver can be designed to exchange data only in a preset frequency band. Despite it defines seven channels of 10MHz bandwidth, the 802.11p standard solves the scan-latency problem restricting the number of channels to discover devices to only one, which is dedicated to periodical dissemination of this control information [9].

Another possible method that could potentially overcome this issue is a similar approach to the one used by the passive RFID solutions. The transceiver does not broadcast periodic frames, but it sends the relevant information for connection after receiving the request from the receiver. A similar method is included in the 802.11 specification with the name of active scanning. Like the single-channel scanning, this service is supported by the *wpa_supplicant* and the analysed drivers, but not by the *Android* API. The problem has been previously studied for a different application in [11]. Unfortunately the proposed solution inevitably means a more complicated installation process, including in some cases root access to the system. These enhancements can discourage potential users from adopting this technology, which goes against one of our main objectives. We should, therefore, accept that the use of an *Android* device as an alternate platform implies a maximum scan delay that is determined by equation (2).

After analysing this limitation we can state that the integrity of the system could be affected by the vehicle speed. The maximum value that guarantees the information prompted before the traffic sign is overtaken can be defined as:

$$s_{max} = R \times t_{max} \quad (3)$$

Although we have initially considered a constant value for R , the experience shows that the maximum distance to establish a safe and durable connection

between the smartphone and the traffic sign is significantly reduced when the experiment is carried out in an urban environment. We have actually moved from 100-meters communication ranges in open spaces, like a highway, to a maximum distance of 40 meters in narrow streets with presence of different obstacles (e.g. tall buildings, parked vehicles, trees, etc.). Despite these limitations, when we consider that the maximum speed allowed in urban areas is 50 km/h, our solution is still able to exchange information before the vehicle overtakes the transceiver. Applying 3 to the highway conditions, the maximum speed obtained is approximately 96 km/h, which means that the reliability of the system can be compromised above that vehicle speed.

On a different theme, we consider that the smartphone is not a dedicated piece of equipment and our solution must avoid any adverse effect on the usual device performance. In order to have a better picture of how our proposal performs on a smartphone, we have run a second experiment with each of the other applications installed on the device and collected the following indicators: CPU usage, memory allocated and battery status. The test consisted of a one-hour trip on an itinerary that was 30% urban and 70% freeway, covering approximately a distance of 50 kilometers. We aimed to establish whether the application could have negative effects on the smartphone performance.

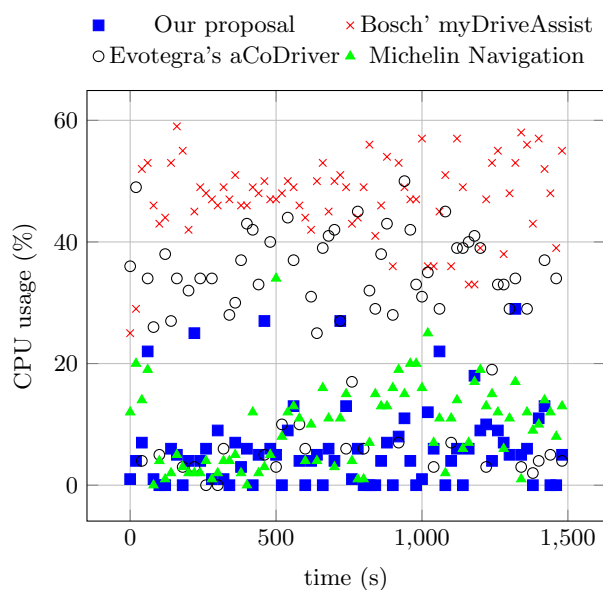


Fig. 7. *GT-I9195* 1.7GHz dual-core CPU utilisation by application.

From the captured data, the two applications that use the camera for sign detection are the ones making the most extensive use of the CPU as we can see

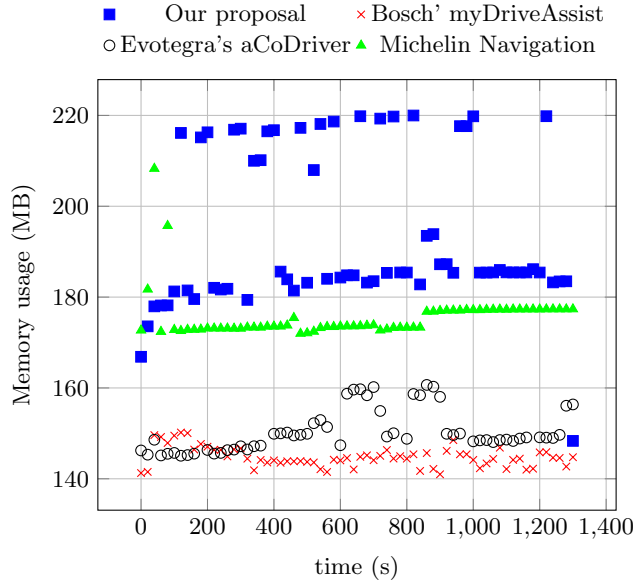


Fig. 8. *GT-I9195* RAM usage by application.

on Fig. 7. The other two, *Michelin Navigation* and our proposal, rely on the network interface as the main input. Unlike the camera, the network adapter is used based on the application demand, and not continuously, hence the CPU load should be in general less uniform, but lower than the camera-based systems.

As a result the battery consumed during the test trial was also more representative on applications that use the in-build camera than on network-based ones. In particular *myDriveAssist* had already consumed 44 percent of the battery charge after one hour, while our application was still showing an 86 percent of the battery available (see Fig. 9). Regarding the memory requirements, on figure 8 the highest figures are registered by our application; however taking the maximum value recorded, it does only represent the 16% of the available RAM memory of the smartphone, and hence we can conclude that it would slightly affect negatively the device performance.

The experiment shows that our proposal of applying wireless technologies to traffic sign recognition does not only offer a more versatile solution than camera or data-map based systems, but also performs better on a resource-constrained device like the smartphone used.

6 Conclusion

In this paper we presented an architecture that aims to facilitate the introduction and deployment of digital sign-posts based on low cost devices and the integration of smartphones as an alternative in-vehicle user-interface. Wireless

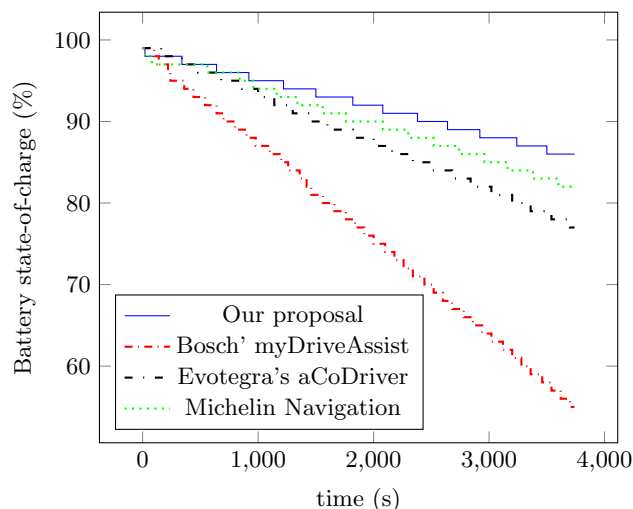


Fig. 9. Real discharge for the Li-Ion 1900mAh removable battery by application.

communications based on IEEE 802.11 is used for the basic connectivity requirements.

Analysis of the data obtained shows that there are major reasons for further studies about the opportunity of using smartphones to expand the number of potential users of this technology. With a low budget we have implemented an application that is able to run for a long time on any *Android* device without potential CPU or memory capacity issues and reliably detect traffic signs equipped with our wireless solution at any vehicle speed up to 96 km/h.

The main limitations we have discovered during the implementation are introduced by the APIs involved in wireless communication for *Android*. The *Wifi-Manager* class, which handles most of the interaction between the application and the Wi-Fi controllers, is very limited compared with other operating systems. Moreover the scanning and association phases are not very well documented in the official *Android* reference, and most of premises we have followed during the implementation are based on our own experience. Nevertheless *Android* is an on-going project and these limitation might be overcome in the future, which would enhance the system response and, therefore, increase the effective maximum vehicle speed.

Despite its current limitations, we have compared our solution with others *Android* applications based on different strategies, and this proposal proves the benefits of using wireless technologies among the other exploit techniques underlined in this document. Our system does not only offer the possibility of detecting a wider range of traffic signs and have additional capabilities like providing live information from the area, but also we have verified that makes a more efficient use of the smartphone resources like battery or CPU.

Acknowledgments

This work was partially supported by the *Ministerio de Economía y Competitividad, Programa Estatal de Investigación, Desarrollo e Innovación Orientada a los Retos de la Sociedad, Proyectos I+D+I 2014*, Spain, under Grant TEC2014-52690-R.

References

1. Alkim, T. P., Bootsma, G., Hoogendoorn, S. P.: Field Operational Test” The Assisted Driver”. In: Intelligent Vehicles Symposium, 2007 IEEE, pp. 1198-1203. IEEE, Istanbul (2007)
2. Trübswetter, N., Bengler, K.: Why Should I Use ADAS? Advanced Driver Assistance Systems and the Elderly: Knowledge, Experience and Usage Barriers. In: Proc. 7th Int. Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design, pp. 495–501. Bolton Landing New York, (2013)
3. Maenpaa, K., Sukuvaara, T., Ylitalo, R., Nurmi, P., Atlaskin, E.: Road Weather Station acting as a wireless service hotspot for vehicles. In: Intelligent Computer Communication and Processing (ICCP), 2013 IEEE International Conference on, pp. 159–162. IEEE, Cluj-Napoca (2013)
4. Djahel, S., Smith, N., Wang, S., Murphy, J.: Reducing emergency services response time in smart cities: An advanced adaptive and fuzzy approach. In: Smart Cities Conference (ISC2), 2015 IEEE First International, pp. 1–8. IEEE, Guadalajara (2015)
5. Yoshimichi, S., Koji, M.: Development and evaluation of in-vehicle signing system utilizing RFID tags as digital traffic signs. In: International Journal of ITS Research 4.1, pp. 53–58. (2006)
6. Pérez, J., Seco, F., Milanés, V., Jiménez, A., Díaz, J. C., De Pedro, T.: An RFID-based intelligent vehicle speed controller using active traffic signals. In: Sensors 10(6), pp. 5872–5887. (2010)
7. Naja, R.: Wireless vehicular networks for car collision avoidance (Vol. 2013). Springer, New York (2013)
8. Huang, W., Zhongdong, Y., Zhu, F., Yang, L., Wang, F. Y.: Applicability of short range wireless networks in V2I applications. In: Intelligent Transportation Systems- (ITSC), 2013 16th International IEEE Conference on, pp 231–236. IEEE, The Hage (2013).
9. ETSI, TCITS. ”Intelligent Transport Systems (ITS); European profile standard on the physical and medium access layer of 5 GHz ITS.” Draft ETSI ES 202.663 (2009): V0.
10. Murray, D., Dixon, M., Koziniec, T.: Scanning delays in 802.11 networks. In: Next Generation Mobile Applications, Services and Technologies, 2007. NGMAST’07. The 2007 International Conference on, pp. 255–260. IEEE, Cardiff (2007)
11. Brouwers, N., Zuniga, M., Langendoen, K.: Incremental wi-fi scanning for energy-efficient localization. In: Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on, pp. 156-162. IEEE, Budapest (2014)
12. Choi, P., Gao, J., Ramanathan, N., Mao, M., Xu, S., Boon, C.C., Fahmy, S.A. Peh, L.S.: A case for leveraging 802.11 p for direct phone-to-phone communications. In: Proceedings of the 2014 international symposium on Low power electronics and design, pp. 207–212. ACM, La Jolla (2014)