

Supporting augmented floating car data through smartphone-based crowd-sensing

Orazio Briante, Claudia Campolo*, Antonio Iera, Antonella Molinaro, Stefano Yuri Paratore, Giuseppe Ruggeri

University Mediterranea of Reggio Calabria, Italy



ARTICLE INFO

Article history:

Received 9 May 2014

Received in revised form 30 July 2014

Accepted 4 August 2014

Available online 19 August 2014

Keywords:

Smartphone

Floating car data

Offloading

ITS

Mobile crowd-sensing

CAN bus

ABSTRACT

Over the last few years, academia, governmental agencies, automotive and electronic industries have increased their interest in Intelligent Transportation Systems (ITS) and services. The leading European ITS stakeholders are pushing to start a few “day-one” applications in 2015. Among them, the collection of Floating Car Data (FCD) generated by moving vehicles is expected to support many practical use cases in a near future, such as road safety provision, in-vehicle diagnostics, and traffic monitoring. This paper shows that *consumer devices on the mass market* (such as smartphones) and *available networking technologies* (such as Wi-Fi and cellular) can compensate for the current lack of dedicated equipment and mature vehicular communication technologies.

The proposed SmartCar (*SMARTphone-based floating CAR data collection*) platform gives a twofold contribution to the efficient support of early ITS applications on a *large-scale* through: (i) an intelligent use of smartphones to collect “augmented” FCD from in-vehicle telematics and external sensors (e.g., pollution detectors for urban sensing), and (ii) the adoption of an *offloading* strategy that leverages Wi-Fi hotspots to alleviate the burden on the cellular network due to the massive generation of “augmented” data. The technological feasibility of SmartCar for augmented FCD gathering and remote transfer is demonstrated through a prototype, preliminarily experimented on board the vehicle and based on *low-cost off-the-shelf hardware* and *open-source software*. Results from preliminary field trials are collected, whereas the effectiveness of offloading at a large-scale is further assessed by a simulation campaign under realistic settings.

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

A wide range of Intelligent Transportation Systems (ITS) services, including safety applications, traffic efficiency, predictive maintenance, infotainment, smart green mobility, and pervasive sensing, can be provided by letting vehicles exchange data with each other and with the road infrastructure.

The following “day-one” vehicular applications have been selected for deployment by the leading European ITS stakeholders starting in 2015: in-vehicle signage, floating car data (FCD), intersection safety/green wave, road works warning, traffic information, strategic routing, hazardous location warning [1]. Among them, especially FCD-related services have the potential to support many practical applications for safety, vehicle diagnostics and road traffic monitoring.

FCD refers to the collection of vehicles' position and kinematics data (e.g., speed, direction of travel) to get traffic information for ITS applications. Furthermore, thanks to the advancements in in-vehicle telematics, FCD can also include reports about the status of microprocessor-based electronic control units networked within the vehicle through the Controller Area Network (CAN) bus. Environmental sensors (like pollution detectors) can be easily brought into the picture to augment the scope of FCD-enabled applications towards urban sensing operations, so that vehicles can be integrated in the future smart-city infrastructure [2]. Henceforth, we will refer to the resulting set of collected data as “*augmented*” FCD.

These augmented data can be processed by remote control centers for multiple purposes, e.g., to monitor road traffic conditions and prevent congestion [3,4], to inform fleets of vehicles cooperatively driving [3], to detect possible in-vehicle malfunctions, to collect traffic statistics, to get car maintenance tips and service information. Depending on the purpose of the FCD collection, data may have different time validity (time interval during which the information is relevant for the service) and accuracy demand. For

* Corresponding author at: University Mediterranea of Reggio Calabria, Loc. Feo di Vito, 89122, Reggio Calabria, Italy. Tel.: +390965875276.

E-mail address: claudia.campolo@unirc.it (C. Campolo).

example, if extracted traffic-related data are used for traffic signs adaptation, then a few seconds validity is reasonable, whereas long-term road traffic statistics may tolerate longer validity times, up to some minutes, and a lower number of samples; finally, data pertaining to long-term car maintenance can be stored for days on the vehicle before transmission.

Despite the wide range of services potentially enabled by *augmented FCD* and the surging interest within standardization bodies, such as 3GPP (Third Generation Partnership Project) [3], ETSI (European Telecommunications Standards Institute) [4], and SAE (Society of Automotive Engineers) International [5], the deployment of standard communication technologies and protocols for the remote transfer of FCD is still underway. Stakeholders worldwide are struggling to finalize standards to allow specialized wireless on-board units (OBUs) to *directly* interact with the CAN bus and to communicate with other OBUs and with roadside units, under any propagation condition, intermittent connectivity, and traffic density [6].

Nonetheless, both technical (harsh vehicular environment) and economic barriers slow down the large-scale deployment and market penetration of Vehicular Ad hoc NETworks (VANETs) and OBU-equipped cars. Such factors challenge the capability of the automotive industry and standardization process to be fast enough in developing and pushing the new technology before the current consumer electronics and network technologies available in the market offer valuable alternative solutions.

It is the authors' convincement that the current need for *augmented FCD* could be satisfied by using *mass-market users' devices*, such as *smartphones*, and *available technologies*, such as *cellular and Wi-Fi networks*. This convincement is shared with the scientific community [7] and many running initiatives are pushing towards enabling users' portable devices, such as tablets and phones, to access in-vehicle telematics [8] and to monitor in-vehicles services through low-cost devices and open-source software [9,10]. In addition, several solutions have been proposed that leverage smartphones in the automotive context, e.g., for safety [11,12] and traffic management purposes [13,14], or to collect driving habits [15–17]. Finally, the Open Automotive Alliance (OOA), a group of leading automotive manufacturers and technology companies [18], also argues for a close alignment of consumer and automotive technologies and pushes towards bringing the *Android* platform to the car, to make in-car technology safer and more intuitive for everyone.

Although the use of smartphones is a timely solution to offer automotive services with a *short-time-to-market*, some issues needs to be addressed when considering FCD-based services. Specifically, the amount of uploaded data from a large number of contributing vehicles with a frequency of seconds risks burdening the cellular network and drain the smartphone's battery. Consequently, either traditional cellular traffic (e.g., voice) could be penalized or FCD applications could achieve poor performance being unable to meet time delivery and integrity requirements. Solutions in the literature, targeting to solve such an issue and alleviate the cellular network load, either leverage the upcoming IEEE 802.11p technology to collect data to be aggregated and remotely delivered [19, 20], or probabilistically select a subset of cars to take part in the process [21] by assuring only partially accurate data retrieval.

The work in this paper addresses the support of FCD services in a more comprehensive way, with the design and demonstration of a solution encompassing smartphones for (*i*) efficient data collection from a wide set of heterogeneous (in-vehicle and external) sensors to effectively provide *augmented FCD services*, and (*ii*) opportunistic data delivery through a simple cellular traffic *offloading* technique designed to exploit on-the-road Wi-Fi connectivity by leveraging the built-in smartphone's Wi-Fi network interface card (Fig. 1).

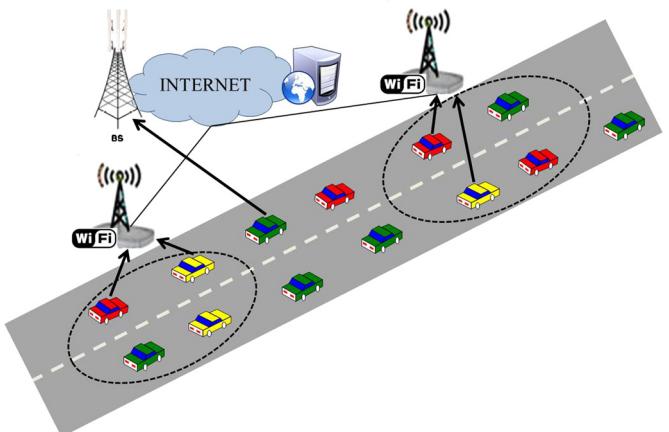


Fig. 1. Reference scenario: vehicles transmit augmented FCD traffic to either the cellular network or a Wi-Fi hotspot.

Cellular network offloading may significantly reduce costs for cellular operators to tackle increasing traffic demands, by avoiding the deployment of additional base stations or the upgrading of the existing ones. Unlike previous literature on offloading techniques, typically focusing on *downlink data transfers* in *low-mobility scenarios* [22], our work addresses *uplink data packets offload in vehicular environments*. Indeed, in [23] vehicular Wi-Fi offloading is preliminarily debated as a promising research direction, provided that some issues uniquely arising in vehicular environment (i.e., short and intermittent connectivity, fast fluctuating wireless channels) are properly overstepped. Some of those concerns may hinder the effectiveness of offloading in the presence of bulky contents [23], but are deadened in case of *small packets*.

An early version of the SmartCar (*SMARTphone-based floating CAR data collection*) design [24] and its preliminary demonstrator [25] are extended in this paper with further functionalities to specifically target *augmented FCD* applications. The prototype deployed on board the vehicle leverages *low-cost off-the-shelf* hardware and *open source* software (e.g., *Arduino* platform [26] and *Android* [27]), which give to the designed solution the potentiality to be an effective mobile crowd-sensing tool.

The rest of the paper is organized as follows. Section 2 describes the set of services enabled by FCD and the issues raised by the evolution towards *participatory FCD*. Section 3 discusses the role of the smartphone in the SmartCar platform, whose main components are presented in Section 4 along with a rough cost analysis. Section 5 discusses the rationale behind design choices of the conceived platform. In Section 6 first field trials are introduced. Section 7 provides quantitative insights into the performance of the offloading technique through simulations conducted under realistic settings for the roads topology, the vehicular mobility patterns, the deployment of Wi-Fi access points, and the smartphone's parameters. Section 8 concludes the paper, providing hints about future work.

2. Floating car data: a day-one ITS application

2.1. FCD services

FCD enables a wide range of services, ranging from *traffic management* to *vehicle diagnosis* and *fleet management*, serving both the individual commuter and the community, and generating potentially high revenues for the stakeholders. In addition, FCD services are among the main representative use cases of a very hot topic, i.e., machine-to-machine (M2M) communications [3,4], considered as the foundation layer for the future world of smart objects, smart homes, smart cities.

Table 1

Towards participatory augmented FCD generation.

Main FCD delivery techniques	Delivery technologies	Involved collectors	Type of collected/inferred data
GPS-based ([1] and references therein, [28])	GPS	Fleets of vehicles only	Position, speed (FCD) Position, speed and other in-vehicle parameters (xFCD) [28]
Phone-based ([1] and references therein)	Passive cellular measurements	Fleets of vehicles only	Position and speed only inferred from network monitoring
Participatory ([19–21,29], SmartCar)	Cellular only [21,29] Hybrid cellular-802.11p [19,20] Offloading (SmartCar)	Potentially all vehicles equipped with a smartphone ([21,29], SmartCar) and with an on board 802.11p unit [19,20]	Position, speed (FCD) [19] Position, speed and other in-vehicle parameters (xFCD) [21,29] Position, speed, other in-vehicle parameters and external sensors measurements (augmented FCD) ([20], SmartCar)

Timing and position coordinates (*timestamp, latitude, longitude, altitude*) of the generating vehicle [1] are the FCD core information elements. They can be enriched to create what is referred to as *extended FCD* (xFCD) [28] with additional information about the vehicle status and kinematics (i.e., driver assistance systems, vehicle light system, engine revolutions per minute, vehicle speed) provided by the CAN bus. The collection of this information may enable applications to monitor traffic conditions, estimate road congestion, and eventually help drivers to avoid areas with traffic jam.

Collected data can also help fleet management to track vehicles and to improve services. For instance, thanks to up-to-date location reports a taxi company can schedule a nearby available taxi to serve passengers so to improve the offered service effectiveness and the customer satisfaction [3]. Remotely uploaded location information can also be used to track freight and stolen vehicles.

Other information retrieved from the CAN bus and concerning the specific vehicle status (e.g., about the anti-lock braking system, the oxygen sensor output voltage, the engine coolant temperature, etc.) may be sent to the car manufacturer and workshops for remote monitoring and diagnosis purposes, for vehicle statistics collection, and also to get tips on maintenance and service information.

2.2. FCD methods: towards a participatory data collection

Traditional FCD methods track vehicles' positions through the Global Positioning System (GPS) or passively through mobile phones. In the former case, positions of taxi and bus fleets equipped with an on-board GPS receiver are regularly tracked by a server. In the latter case, the mobile phone position is inferred by the cellular network operator by relying on triangulation measurements (if possible) or on handover events. With such an approach, the position estimation is less accurate than with the GPS-based solution, and it is only feasible in case of overlapping cell coverage, such as at those road segments that represent a border line between two (or more) neighboring cell areas where handovers may occur [1].

The new recent trend in the FCD collection technique is to actively involve *each vehicle* in the FCD generation and transmission, instead of delegating the network to retrieve and infer the vehicle position. In this *participatory* FCD concept, data retrieved by the vehicles are regularly transmitted to a remote control centre through cellular technologies. Promoting participatory data collection on a *large-scale*, by also involving private commuters besides fleets, is particularly helpful when the infrastructure (e.g., inductive loops) is either unavailable or insufficient to provide accurate predictions

and measurements, or too expensive to install and maintain. Table 1 summarizes some examples that testify to the paradigm shift from early FCD applications to recent *participatory* approaches; it allows to better figure out where our SmartCar proposal is placed in this context.

2.3. FCD load and proposed solutions

Several vehicles collecting and transmitting FCD could bring about concerns about the excessive data upload traffic burdening the network. Despite the small data size, up to nearly 100 bytes¹ per packet, the data generation rate (a FCD report may be transmitted every 1–10 s) and the number of involved vehicles (it could be hundreds in large cells) can *heavily burden the uplink channel of the cellular network*, thereby threatening the delivery of traditional traffic or hindering the timely FCD delivery.

Recently, some techniques have been proposed to alleviate the load of FCD traffic on the mobile data network. In [21,29] vehicles probabilistically transmit FCD over Long Term Evolution (LTE), based on the measured signal-to-noise ratio (SNR): the higher the SNR, the higher the FCD transmission probability. The channel load is consequently reduced, but the number and distribution of cars generating FCD do not always assure reliable and accurate traffic forecasts.

A hybrid *clustering* technique is proposed in [19] that relies on LTE and IEEE 802.11p technologies. The eNodeB is responsible for the formation of clusters of vehicles, communicating over 802.11p, and the selection of the relevant cluster heads (CHs). Each CH transmits XFCD received from its cluster's members to the eNodeB. Thanks to the aggregation performed by the CH, the proposed technique provides significant advantages in terms of bandwidth usage and packet delivery. With similar purposes, in [20] a subset of vehicles gathers and aggregates the data sensed by neighboring cars through short-range communication, and uploads the aggregated information via a single packet transmission. The solutions in [19] and [20] meet the objective of reducing the load of the uplink of the cellular network, but, these rely on IEEE 802.11p-enabled on-board devices, whose penetration rate is expected to increase at a slow pace and whose cost may be very high at the initial deployment stage.² Our proposal, instead, appears as a promising *cheap and short-time-to-market solution* that, by leveraging widely

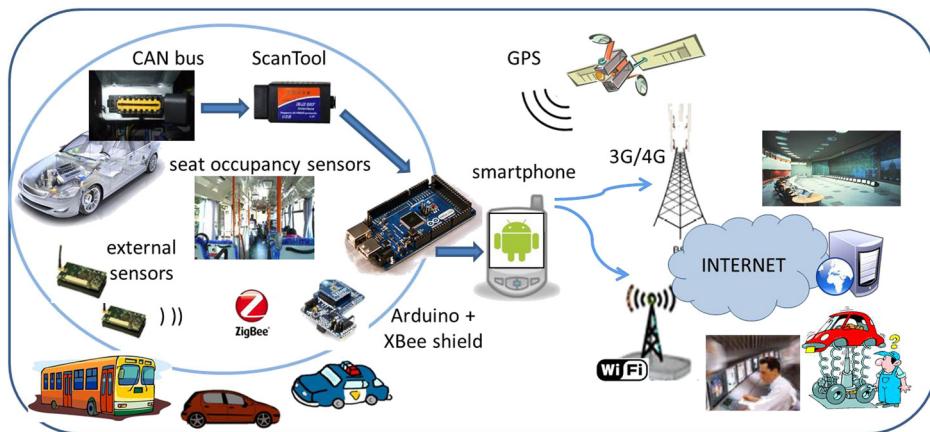
¹ The minimum data footprint per packet is around 25 bytes, for timestamp, position, speed and direction information [30] and may increase according to the number of additional conveyed elements concerning vehicle status and its surroundings.

² The cost of commercial OBUs compliant with IEEE 802.11p specifications, like the ones by Cohda Wireless [31] is around 3000 €, whereas for a 802.11p PMcia [32] mounted into an Alix board it may be up to 500 €.

Table 2

Main representative solutions for participatory FCD collection.

Reference	Communication technologies	Collection mode	On board equipment	Energy saving issues	Market penetration rate	Cost for devices and data delivery
[21,29]	LTE only (no Wi-Fi offloading)	Probabilistic (SNR-based)	LTE user equipment (e.g., a smartphone)	Addressed through limitation of transmitted FCD (based on channel quality)	Potentially high	Affordable cost of the device + LTE connection fee
[19,20]	802.11p to gather data in the local neighborhood; cellular [20]/LTE [19] technology for remote aggregated data delivery	Clustering	802.11p + cellular [20]/LTE [19] interfaces	Not an issue (on-board devices powered by the vehicle)	Low and increasing at a slow pace	Medium-to-high cost of dedicated embedded devices + cellular/LTE connection fee
SmartCar	Cellular and Wi-Fi for remote data delivery	Offloading	Smartphone, Arduino board, OBD connector, USB cable, sensors	Not an issue (smartphone powered through the vehicle's battery)	Potentially high	Platform cost estimation + cellular connection fee (in Section 4.5)

**Fig. 2.** The central role of smartphone in the *SmartCar* platform.

available communication technologies, easily achieves a *large-scale participation of commuters*.

Unlike the proposals in [19] and [21], our solution extends the scope of FCD by effectively exploiting the car as a powerful multi-sensor platform, also able to gather city-wide environmental measurements while moving. Besides the data delivery strategy, this paper also describes the hardware and software components deployed to retrieve data from the in-vehicle telematics and other sensors, thus providing a comprehensive framework for supporting *augmented FCD collection and delivery*.

Table 2 provides a rapid qualitative comparison between SmartCar and the current solutions for participatory FCD collection. We classify them into different typologies, based on the radio technologies (some of them use LTE only, others use a combination of LTE and Wi-Fi/IEEE 802.11p), the collection mode (e.g., probabilistic, cluster-based, offloading), the equipment on board, the expected market penetration and costs for the devices and the data delivery.

3. The central role of the smartphone

The smartphone plays a central role in our proposed solution and implements most of the required tasks, ranging from data collection and pre-processing to remote delivery. A functional scheme of our SmartCar platform is reported in Fig. 2.

The smartphone simultaneously collects information provided by *multiple heterogeneous sources*:

- *The embedded GPS receiver.* FCD should at least include timing and position coordinates of the vehicle; this information is directly taken from the smartphone that is endowed with self-positioning capability.³
- *The in-vehicle CAN bus.* In addition to kinematics-related data, information concerning the specific vehicle status (e.g., engine status) are also retrieved from the CAN bus.
- *Seat-occupancy sensors.* This feature is an add-on with respect to the platform described in [24]. Detecting and remotely transmitting the seat occupancy status can serve several purposes, e.g., facilitating the planning of rescue operations from remote centers. As a matter of fact, ETSI hazard notifications messages [33] also carry this information. Moreover, such an information may provide statistics about passengers in public vehicles (like taxi, bus, mini-bus) for ticketing and service planning, hence of particular interest for *fleet management applications*.
- *Environmental sensors.* For example, in addition to traditional FCD retrieved from the in-vehicle telematics, air quality measurements gathered through cheap sensors mounted on the vehicle avoid the deployment of sophisticated sensors or pollution monitoring stations on the roads.

³ Such a feature can be used in any type of cars, regardless whether they are equipped with a GPS receiver or not.

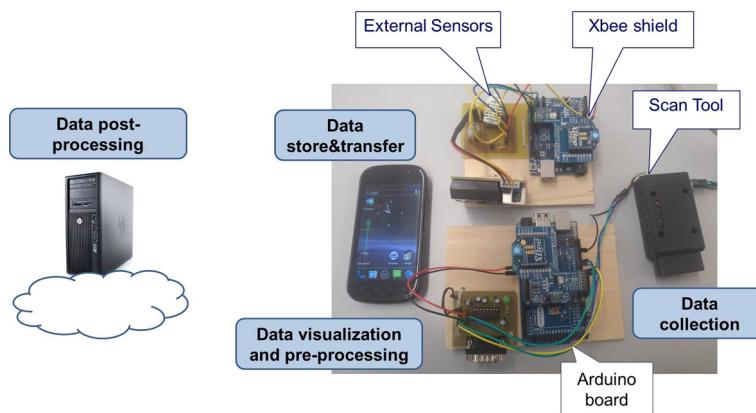


Fig. 3. Modules running over the components of the SmartCar platform.

The smartphone is also responsible for delivering the retrieved data to remote processing centers, through its built-in cellular or Wi-Fi radio interfaces. In order to offload the cellular network, the smartphone is augmented with a *software module* (*Data Store & Transfer*) that chooses the interface to transmit the collected data, as detailed in the following section.

4. The SmartCar architecture and functional modules

The *SmartCar* prototype (i) collects in-vehicle FCD and environmental sensing information from outside, and (ii) opportunistically transfers them to meet their delivery constraints and *offload* the cellular network. The platform includes the following functional modules running over the main SmartCar components as shown in Fig. 3:

- *Data Collection* module, responsible for collecting, merging and temporarily storing data. It is implemented in a dedicated USB-enabled microcontroller board receiving data inputs from a variety of sensors, on board the vehicle and in its close surroundings. The open-source, low-cost, flexible hardware/software *Arduino* platform [26] is used to this purpose. It can be configured to offer a variety of wired and wireless communication interfaces. A customized homemade microprocessor could also replace *Arduino*.
- *Data Visualization and Pre-processing* module, implemented in the smartphone. It is independent of the kind of collected data, the type of sensors, and the CAN bus network. It hides to the end-users details about implementation and sensing devices. The smartphone application retrieves data from the *Arduino* board and shows them after processing on a graphical user interface (GUI).
- *Data Store & Transfer* module, implemented in the smartphone, remotely transmits collected augmented FCD according to the data validity, the available network coverage and radio interfaces. It implements the *offloading* strategy.
- *Data Post-processing* module, implemented in the remote server to track vehicles and store the augmented data sent by the smartphone.

A mobile app has been implemented that runs in the smartphone using the Android open source operating system [27]. The software includes the following main functions (i) retrieving data from the *Arduino* board, (ii) tagging data, (iii) showing data to the end-user through a GUI, and (iv) remotely transfer them. Functions (i)–(iii) are performed by the *Data Visualization and Pre-processing* module; function (iv) is performed by the *Data Store & Transfer* module.

4.1. The data collection module

4.1.1. In-vehicle telematics

Data supplied from the vehicle's CAN bus can be read through the On-board Diagnostics (OBD)-II interface. OBD-II is the standard that specifies the type of diagnostic connector and its pin out, the electrical signaling protocols, and the messaging format [34]. Specifically, it provides both real-time vehicle data and a standardized series of Diagnostic Trouble Codes (DTCs), allowing to quickly detect and solve malfunctions in the vehicle.

The OBD-II interface is not available in a smartphone, rather it can be provided by an OBD connector. An ELM327 chipped *scan tool* has been used to this purpose; it includes a controller, a CAN transceiver, and an OBD male connector; it is compliant with the OBD-II specifications.

Several ready-to-use OBD connectors are available on the market, either endowed with USB interface or providing Bluetooth and Wi-Fi connectivity to directly interact with smartphones.

In our prototype, an *Arduino* board is included between the OBD connector and the smartphone, instead of enabling direct communications over the radio interface.

In particular, the ELM327 acts as a bridge between the OBD ports and a standard RS232 interface. Through the latter one, it is connected to the *Arduino* board over which *AT commands* are sent.

The designed module can potentially retrieve the full spectrum of standardized in-vehicle data. Notwithstanding, a software module, loaded into the *Arduino* platform, filters data from the CAN bus to avoid polling a high number of in-vehicle sensors, which would increase the contention on the CAN bus and reduce the data sampling rate, as demonstrated in [12]. The sampling rate can be set according to the application requirements.

OBD-II parameter identifiers (PIDs) are used to request a specific vehicle parameter from the CAN bus. The scan tool, triggered by *Arduino*, sends a PID to the vehicle's bus. The device responsible for that PID reports the corresponding value to the bus; then the scan tool reads the response and replies to *Arduino*. Some PIDs like the vehicle speed are closely related to traffic management purposes; while others give information about the vehicle status and enable the car diagnosis from a remote site, to detect possible malfunctions and to collect vehicle statistics.

The *Arduino* is interfaced with an Android smartphone. Such an interface is deployed through the Accessory Development Kit (ADK) [35], a standard powered by Google to enable communication between Android devices and external hardware (in this case, the *Arduino* microcontroller board). In particular, an *Arduino* Mega ADK microcontroller has been used and equipped with a Secure Digital (SD) memory card to temporarily store data. It has a USB host interface to connect with Android-based phones. Smartphones with new Android releases natively support ADK.

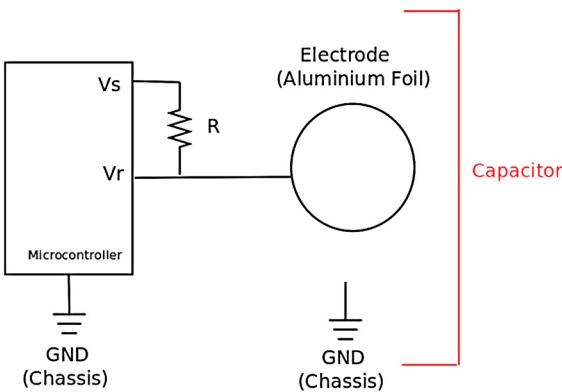


Fig. 4. The scheme of the seat occupancy sensors.

4.1.2. Add-on sensors

Currently, in addition to the CAN-bus, our prototype foresees the interaction with (i) homemade low-cost seat-mounted proximity sensors located in the vehicle to monitor the passengers load, and (ii) environmental sensors located outside the vehicle for monitoring purposes.

The seat occupancy sensors are based on capacitive effects [36]. The set-up scheme relevant to seat occupancy is shown in Fig. 4.

An electrode is placed on the seat to be monitored. This electrode is built up from a thin circular metal plate. In its cheapest realization it could be obtained from an aluminum foil. The electrode is covered by a piece of non-conducting material which can be the seat upholstery itself or a paper covering. The electrode constitutes one plate of a capacitor and the chassis of the vehicle constitutes the second plate of this capacitor. When a human body occupies the seat, the electrical properties between the sensor electrode and the ground change and the total capacitance of the system above described changes accordingly from a value C_{unocc} to a value C_{occ} .

The Arduino board periodically checks the capacity of the system by using two of its pins which in Fig. 4 are labelled as V_s and V_r . The estimation of the capacity is carried out indirectly by measuring the time constant of the resistor-capacitance circuit constituted by the resistance R , which in our set up has a constant value of $10\text{ M}\Omega$.

In [36] the relationship between C_{unocc} and C_{occ} has been modeled both analytically and empirically and the following relation has been determined: $C_{occ} = 2.05C_{unocc}$. In our set-up we conclude that a seat is occupied if the capacity measured exceeds $1.7C_{unocc}$. More details about the seat occupancy sensors can be found in [36].

The environmental sensors transfer measured data to the Arduino board, which has been augmented with XBee modules to support ZigBee communication [37]. The Arduino board can retrieve environment-related parameters at the same or at a different frequency from the one used to get data from the CAN bus. Further sensors can be easily integrated, by adding proper shields to the Arduino board (e.g., biometric data can be collected from drivers and passengers to give more information to rescue teams in case of car accidents).

The augmented FCD information collected from the mentioned set of heterogeneous sensors are packed together and passed to the smartphone.

4.2. The data visualization and pre-processing module

The information retrieved from Arduino is complemented with information gathered by the mobile phone itself, i.e., GPS information, time and position coordinates. Although the type of collected



Fig. 5. A snapshot of the Android-based SmartCar app showing some OBD parameters.

data depends on the remote application, the user is allowed to select the parameters to be directly visualized on her smartphone. The parameters of the selected sensors are then shown in the main window of the GUI and periodically refreshed to give the user a feedback about the car performance. A sample snapshot of the application visualizing some of the in-vehicle monitored parameters, is shown in Fig. 5.

4.3. The data store and transfer module

Collected data, tagged by the *Visualization and Pre-processing module*, are separately queued, based on the priority and lifetime set by the application, and then transmitted to a remote server. Cellular and Wi-Fi interfaces are alternatively used to this purpose. In SmartCar, available connectivity options are monitored in the device by leveraging off-the-shelf Android utilities. Our solution avoids the energy consumption related to continuous scanning for nearby APs [23], since the smartphone is powered by the vehicle's battery (as it would be clearer in Section 5). The decision on the network interface to be used for the remote delivery is based on *opportunistic offloading* principles.

Similarly to the solution in [22], data are always exchanged (uploaded in our case) via Wi-Fi whenever an access point (AP) is detected. Otherwise, when the cellular connection is only available, the decision whether to transmit data or not is taken according to the data *lifetime* associated to each packet (Time-to-Live, TTL).

In particular, we identified three data types: *high-priority*, *medium priority*, and *low-priority delay-tolerant* data. High-priority data, reporting information about a sudden vehicle fault, typically have a lifetime in the order of milliseconds and are *immediately* transmitted over the most reliable and lowest-latency radio interface; the message could be enriched with information about the number of passengers detected by the seat occupancy sensors.

The lifetime for medium priority data may vary according to the application processing them. For instance, it can range from a few seconds, if extracted traffic-related data are used for traffic signs adaptation, to some minutes for long-term road traffic statistics. Moreover, the TTL can vary according to the time of the day and the vehicle speed. For instance, during rush hours, the TTL should be short in order to provide up-to-date information on the traffic situation; on the other hand, data received at night can be considered valid for longer as it is unlikely that the traffic situation will abruptly change [13].

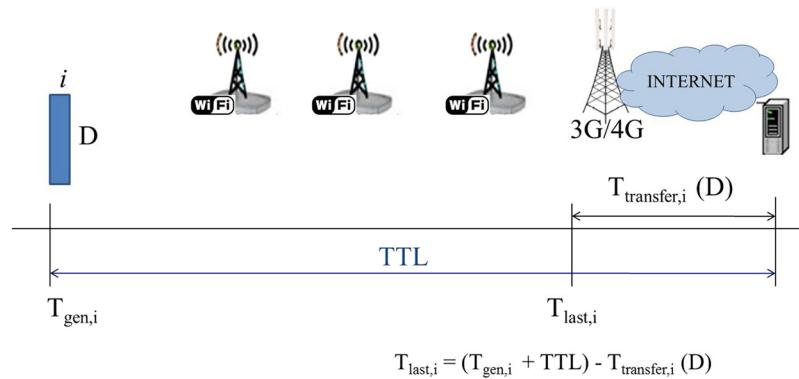


Fig. 6. Main time parameters in the offloading algorithm.

Delay-tolerant data, such as the Diagnostic Trouble Codes concerning the specific vehicle status, to be sent to car manufacturers and workshops for monitoring and diagnosis purposes, can be stored in the smartphone until the vehicle falls under the coverage of a Wi-Fi network or the user owning the device reaches her home/office network.

Overall, medium and low priority packets can be stored in the smartphone until their lifetime expires. Each packet of size D is labeled at the smartphone with a *timestamp* representing its generation time (T_{gen}); the packet expiry time is fixed to ($T_{gen} + TTL$). T_{last} is set as the last time instant that the smartphone can wait before transmitting the packet on the cellular network. If a Wi-Fi hotspot is detected before T_{last} then the packet is transmitted to the AP, otherwise the packet is transmitted through the cellular network. T_{last} is computed as the difference between the packet expiry time and the end-to-end packet transfer time $T_{transfer}(D)$. The latter parameter includes the delay across the cellular network (both radio access and core network segments) and across the Internet to the remote server. For the sake of clarity, time parameters are depicted in Fig. 6.

4.4. The data post-processing module

Data delivered to the remote server can be stored, processed, and analyzed by different applications. Anonymous information are collected; data about the vehicle and the driver are not transferred to prevent privacy violation. The anonymization method in [38] can be used to this purpose. Data collected by SmartCar-equipped vehicles over a large scale can be used to feed providers' servers that monitor the actual traffic situation in real-time (e.g., speeds, volumes) and offer accurate and reliable information both to drivers and road authorities. How such procedures are managed is outside the scope of the present work.

4.5. Rough cost analysis

For the sake of completeness, a rough estimation of costs for the SmartCar prototype is summarized in Table 3.

Concerning the hardware components, the smartphone, which represents the core of the conceived platform, has the highest cost. However, it is highly likely that users already own a smartphone and only needs to upload the SmartCar app.

The costs of other required hardware modules, i.e., the Scan Tool, the Arduino board, and additional shields and sensors, will not be higher than 100 €.

The connection fee depends on the type of charging contract with the network operator; these costs can be considered negligible compared to the hardware costs. By assuming an average car use per day of 1 h, the vehicle could generate nearly 3 MB per day, and 90 MB per month. In case of a flat Internet rate charge

Table 3
Cost estimation.

Category	Description	Cost
Hardware	Smartphone	≥100 €
	ScanTool	≈20 €
	Arduino board	≈40 €
	Sensor (e.g. CO ₂)	<5 €
	XBee shield	≈20 €
Connection fees	Cost per transferred MB	1.40 cent

(typically 10–20 € per month), this amount is much lower than the usual allowed monthly data transfer (e.g., 500 MB or more). In case of a *pay-per-use data charge*, the data transfer cost may range from 3 cents to 120 cents per day. Such a cost can be anyway reduced when offloading to a free Wi-Fi hotspot.

In summary, the overall low cost of the SmartCar platform allows us to argue in favor of its viability and deployment on a large scale.

5. Design rationale

The design of SmartCar carefully considered the following objectives: (i) keeping the smartphone battery drain low, (ii) retrieving a large set of heterogeneous data, (iii) providing a low-cost and user-friendly solution.

The collection and delivery of augmented FCD involves power-hungry tasks that contribute to the smartphone battery consumption (e.g., data collection and transmission, GPS readings, CPU operations, data display). Among them, the major consumption is related to the radio frequency (RF) front-end. Table 4 reports power consumption measurements associated to the progressive activation of the radio interfaces on the SmartCar prototype. Measurements are collected by powering the prototype through a stabilized power supplier and periodically measuring the current flowing in the phone through a 5 and 1/2 digital multimeter. A dedicated LabVIEW program is used to this purpose.

The estimated battery duration has been computed by considering nominal battery's values of real smartphones (1500 mAh, 3.7 V). No aging or leakage effect has been considered in the battery model, so the expected lifetime in Table 4 is a little bit overestimated, especially in low consumption mode. A dramatic reduction in the battery lifetime (it falls to about 3 h) is evident when all the communication interfaces are active. And this figure decreases even more when considering the data transmission over the active radio interface (cellular and Wi-Fi) [39] and the GPS fixing [40].

The mentioned consumption figures indicate that collecting and delivering augmented FCD can be energy-consuming for a battery-powered smartphone. This is the reason why we place an Arduino

Table 4

Power consumption of the prototype when activating radio communication interfaces and estimated battery lifetime.

Operating mode	Consumption [W]	Estimated battery lifetime [hour] (1500 mAh, 3.7 V)
No wireless interface active	0.027	197.62
UMTS interface active; no data transmission	0.053	102.79
UMTS + Wi-Fi active but not connected; no data transmission	0.066	42.19
UMTS + Wi-Fi connected via TLS (Transport Layer Security) authentication; no data transmission	1.503	3.53
UMTS + Wi-Fi connected TLS Authentication + GPS active; no data transmission; no GPS fixing	1.665	3.14

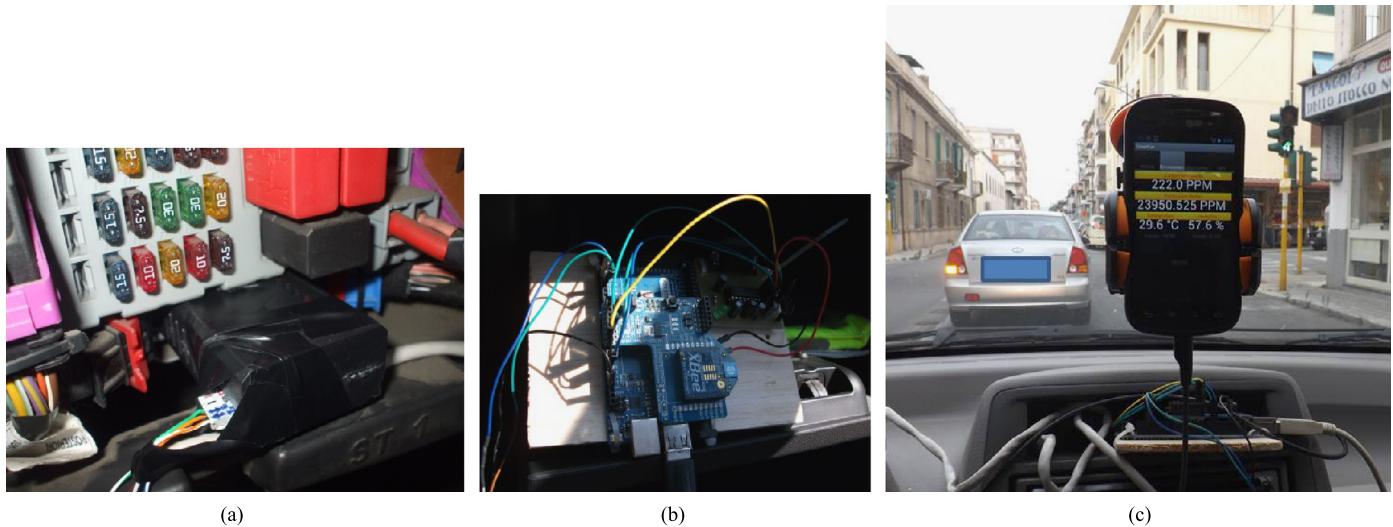


Fig. 7. SmartCar system deployment: connection to OBD II interface (a), the Arduino board (b), the smartphone in the windshield with the GUI showing collected environmental parameters (c). The set-up of the built prototype can be found in the video at this link: <http://youtu.be/ysy70lycd9E>.

board between the OBD-II interface and the smartphone with the twofold role of alleviating the computational load on the phone and to power it. Specifically:

1. We move many power-consuming tasks to the Arduino board. Arduino is in charge of collecting and pre-processing heterogeneous data from the OBD interface, from the in-vehicle and external sensors. Arduino may collect and store data even when the smartphone is absent or switched off, indeed acting as a *black box recorder*. The data accumulated in the Arduino board will be transmitted to the smartphone as soon as it is available/active again.
2. The Arduino microcontroller is directly powered by the vehicle which it is connected to, and is used to give power to the smartphone via a USB cable. The Arduino's USB port is configured in *host* mode so to power the phone through the same USB cable used to transfer the data.⁴ Connecting the smartphone to the USB can be facilitated in the future by embedding the USB plug into the cell phone mount. Therefore, we are confident that the need to use the USB connection will not be too annoying for the user.
3. The Wi-Fi interface card of the smartphone is dedicated to offloading; while other available wireless interfaces (e.g., Bluetooth) are not used to manage interaction between the phone and the OBD-II or the Arduino and are left available for other

usages (e.g., headset for hand-free interaction with the cell phone, infotainments commands, etc.). By doing so, the smartphone will not be engaged in polling/receiving data (from the CAN-bus or from the Arduino board) through Bluetooth/Wi-Fi, and its energy will be further saved.

6. SmartCar: experimental evaluation

We experimentally evaluated the SmartCar platform by performing preliminary field trials on a *single-vehicle*. We plugged the ScanTool into the OBD-II interface of our testing vehicle. The ScanTool has been connected to the Arduino board, and a *Samsung Nexus A* smartphone has been connected to the Arduino board via the USB interface. The smartphone with built-in ADK support and augmented with the implemented *SmartCar* app, was mounted in the windshield of the vehicle. Humidity, temperature, and carbon monoxide sensors have been integrated into the Arduino placed on the dashboard.⁵ Some photos taken during the setup of the prototype and the experiments are shown in Fig. 7.

The experiment has been carried out as follows: we drove the equipped car from the University campus to the Reggio Calabria city center, travelling along the waterfront during the morning rush hours, and driving back to the starting point through a different road. The augmented FCD supplied by the SmartCar-equipped

⁴ This in-vehicle powering system was successfully tested with many modern smartphones (including Android, iPhone and Windows Phones). Although there are still few old phones that cannot be powered through a USB port, we guess that smartphones equipped with a USB port will be more and more pervasive on the mass market.

⁵ The location of environmental sensors in the platform, either outside or inside the vehicle, is expected to affect the accuracy of measurements. Results are available in the literature that show correlation among inside and outside measures [41]. However, at this stage of research, focus is on the viability of the overall proposed solution. Improvements in the accuracy and the deployment options will be a subject matter of future work.



Fig. 8. A server-side view of the vehicle tracking during an experiment in the city of Reggio Calabria (“vehicles” markers represent data remotely delivered over Wi-Fi).

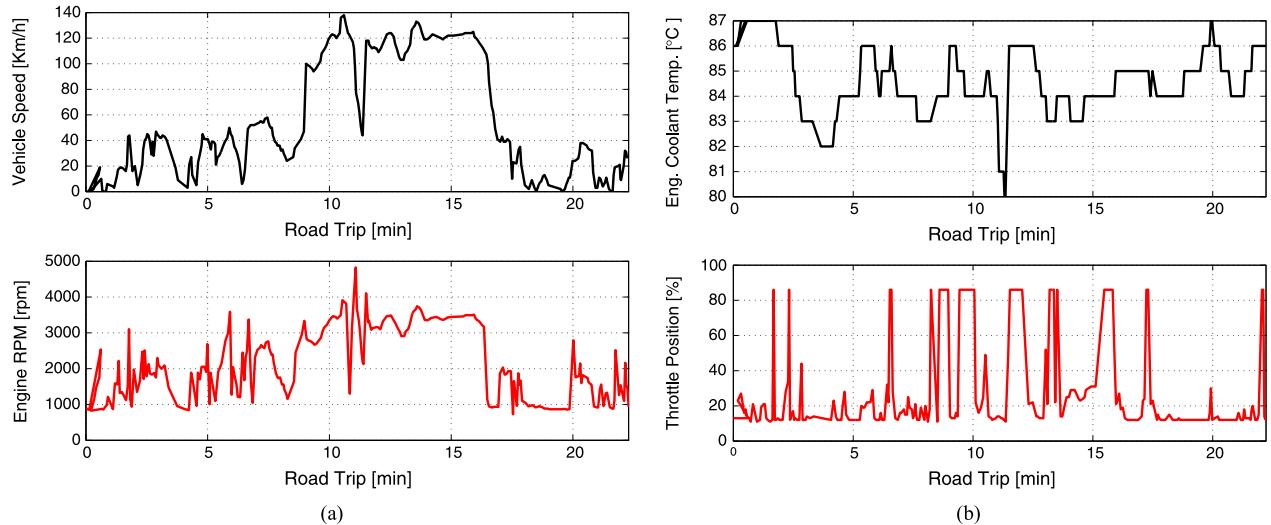


Fig. 9. Telemetry of the SmartCar-equipped vehicle recorded during the test-drive.

vehicle were transmitted by the smartphone through the APs detected on the road.

The SmartCar server-side component, located at the facilities of the Telecommunications laboratory, followed the route of the testing car on a map and displayed the received measurements. Fig. 8 shows a snapshot of the map that graphically illustrates the positions of data offloading during the car journey. The majority of “marker” vehicles that offload data through Wi-Fi are found on the waterfront (left side in Fig. 8), where several free-access APs are deployed.

Fig. 9 shows some telemetry data collected from the SmartCar-equipped vehicle and post-processed at the server-side. Specifically, the speed, the engine revolutions per minute (RPM), the throttle position, and the coolant temperature are reported, as sampled each second from the CAN bus. Those data can help, for example, to infer information about the driving habits of a specific driver (e.g., how safe she drives, how frequently and gently she pushes the accelerator pedal). Each collected sample is associated with the position where it was taken. As depicted in Figs. 10(a) and 10(b), the vehicle trip can be shown on a map through the

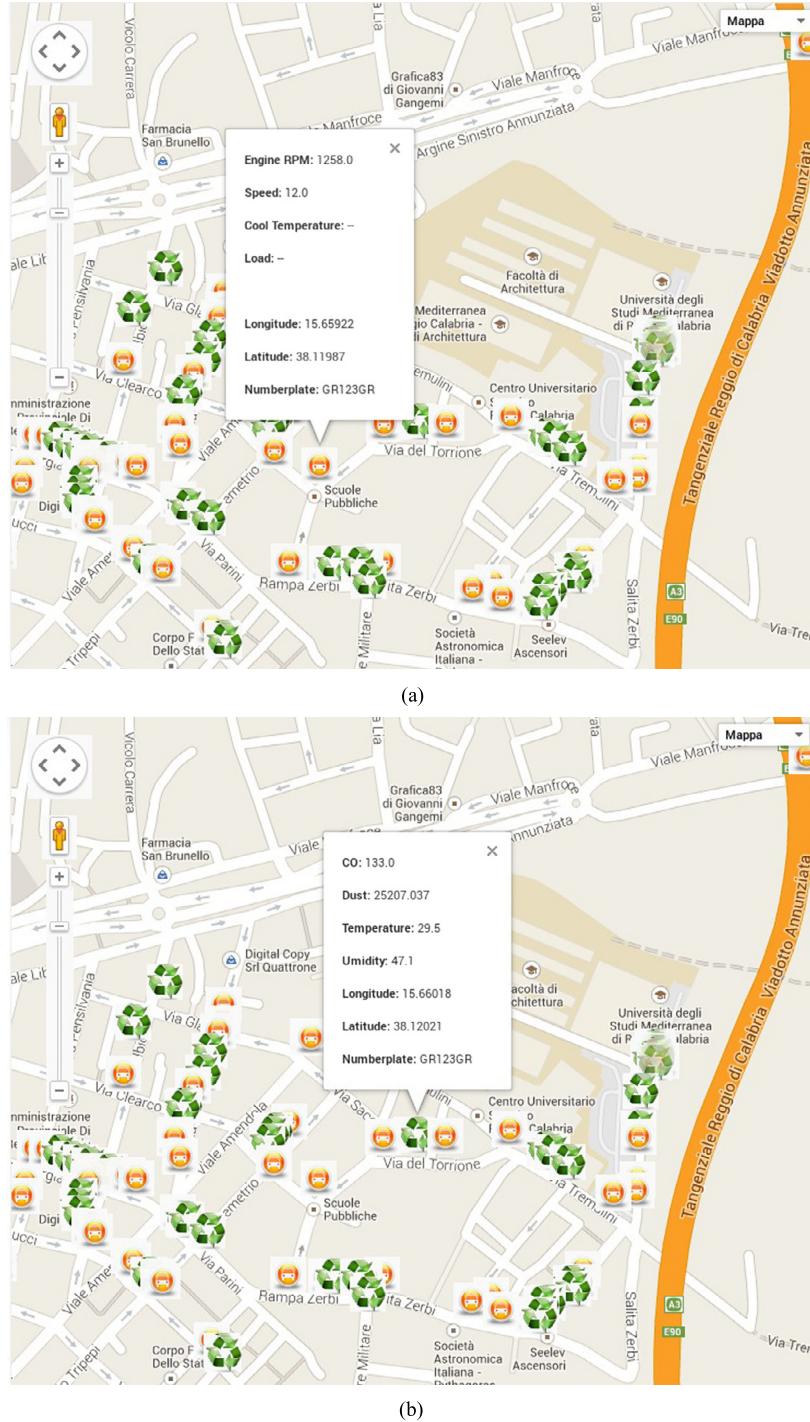


Fig. 10. Visual tracking of a vehicle on a server-side map augmented with OBD-data (a) and environmental measurements (b).

SmartCar server with the relevant data collected by the vehicle from the OBD and the environment.

More sophisticated graphical tools can be used for deeper analyses. For instance, in Fig. 11(a), the carbon monoxide values measured during the trip (illustrated in Fig. 11(c)) are shown on a map and can be used for a quick inspection of the pollution conditions in different areas of a city. Fig. 11(b) reports the vehicle speed as sampled each second at different locations; this information could be used, e.g., to infer the location of traffic jams.

Plots and snapshots in Figs. 9–11 serve only the purpose to show exemplary processing activities that can be performed on the

data collected by our platform; a more sophisticated data analysis is outside the scope of the paper and are left as a task of providers and authorities, which such data are sent to.

7. Analysis of the offloading strategy

Due the prohibitive costs of deploying field tests involving several equipped vehicles, simulations using the Network Simulator (ns-2) [42] have been run to evaluate the offloading performance at a large scale. Tests have been carried out under realistic topol-

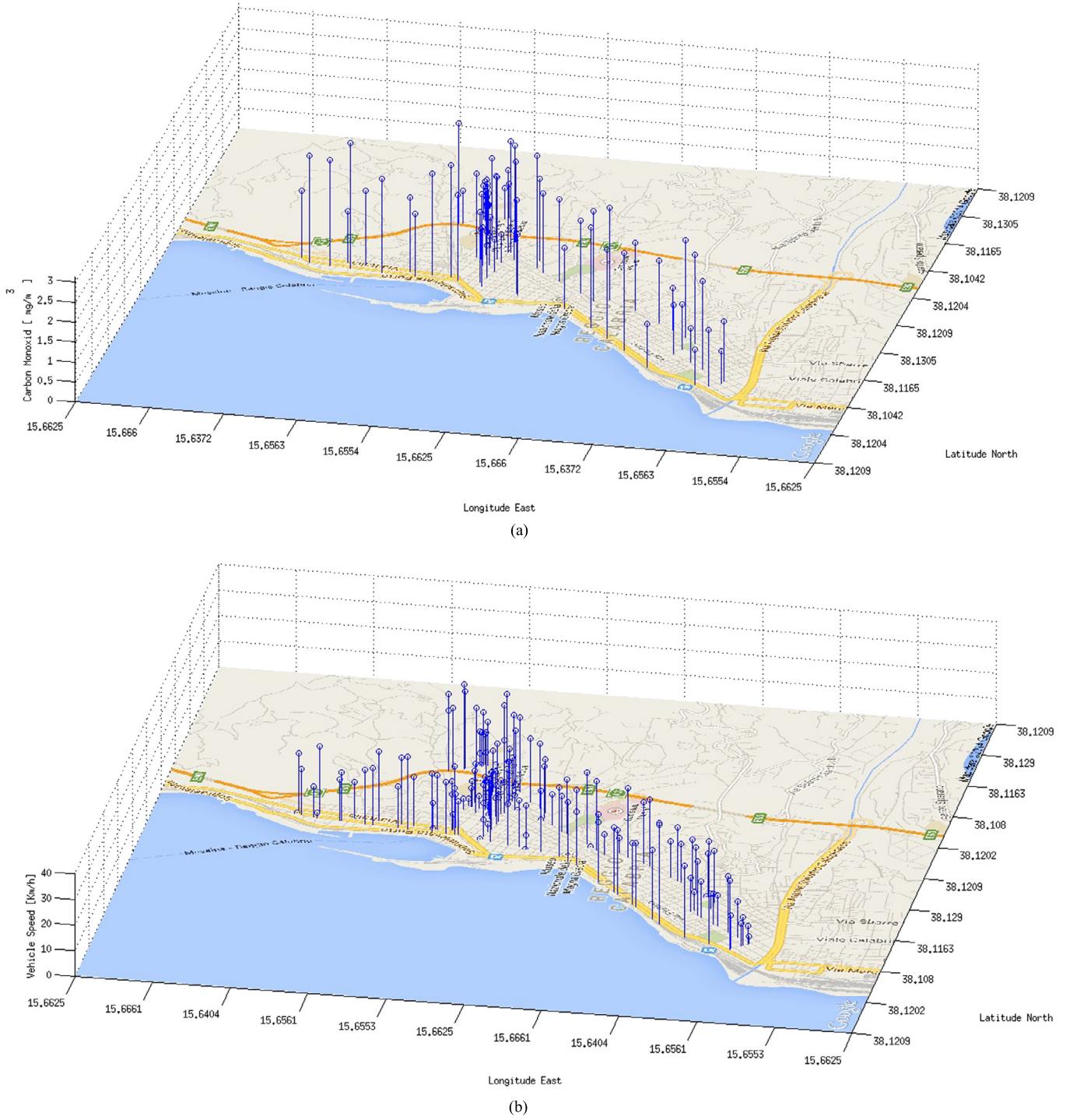


Fig. 11. Carbon monoxide values (a) and vehicle speed values (b) measured during the performed trip whose trajectory is shown in (c).

ogy, mobility, device features, LTE and Wi-Fi APs deployment settings, as summarized in Table 5.

7.1. Simulation settings and assumptions

To create a realistic road layout we used maps publicly available from the OpenStreetMap (OSM) project [43] and combined them with realistic vehicle mobility traces derived from SUMO [44]. Three maps of different zones in the city of Rome have been chosen; each map of a size of nearly 3.2×1.7 km, as illustrated in Fig. 12. The positions of public Wi-Fi APs deployed in the three areas are available from [45], and reported in Fig. 13.

The mobility traces of 100 vehicles are given as an input to ns-2 together with the APs' coordinates, available in latitude and longitude converted in 2D positions.

The APs regularly send beacon frames with a frequency of 10 Hz. Smartphones on board of vehicles generate 100 bytes-long augmented FCD packets that include position, timing, kinematics and environment-related data. They are generated with a frequency of 1 Hz while vehicles move in the topology. As soon as a smartphone detects the presence of a nearby AP, by receiving a beacon, it can connect to the AP and transmit its data. Unless differently stated in the text, the connection time is set to 4 s to account for association and authentication procedures [46].

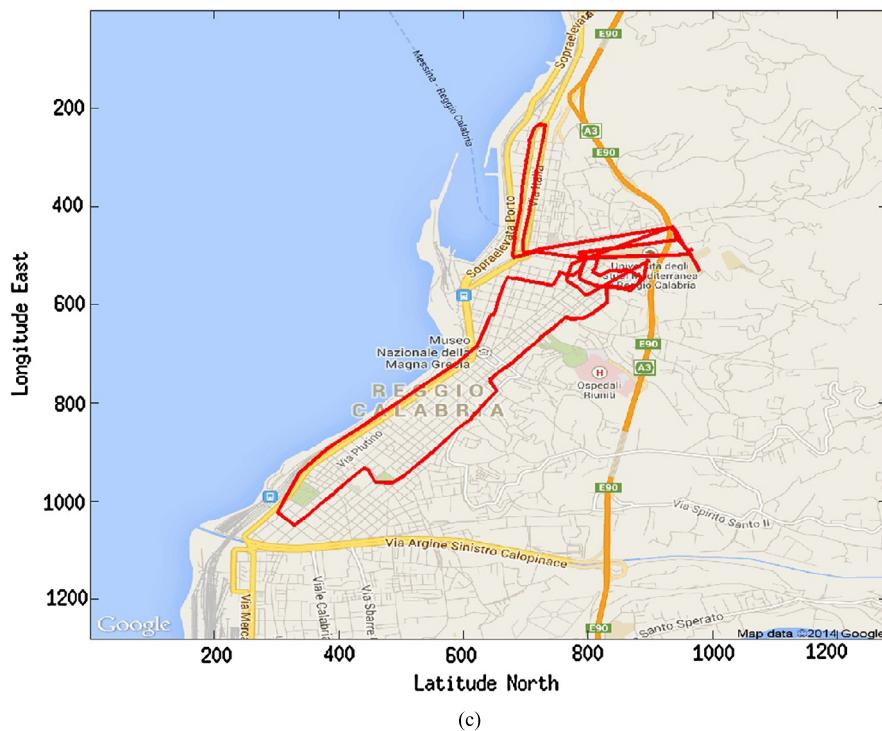


Fig. 11. (Continued.)

Table 5
Main simulation parameters.

Category	Parameter	Value
Application layer	Augmented FCD packet size and frequency	100 bytes 1 Hz
Mobility and topology	Number of vehicles per map zone and in the LTE cell Maximum vehicle speed Number of APs	100 (50, 75, 100, 125, 150) 60 km/h 11 (suburban area); 55 (close to center); 97 (city center)
Access technology: Wi-Fi	Physical/MAC Layer Propagation model Beacon frequency Smartphone's transmission power P_{tx} Smartphone's receiver sensitivity ^a R_s	IEEE 802.11g Nakagami $m = 1$ 10 Hz 16 dBm (type 1), 13 dBm (type 2) −66 dBm (type 1), −74 dBm (type 2)
Access technology: LTE	Available Uplink Resource Blocks (RBs) LTE MAC layer scheduler Transmission power of the LTE user equipment and the eNodeB	25 Round robin 20 dBm (UE) 40 dBm (eNodeB)

^a It is the minimum input power level at which packet decoding can be successfully achieved.

Smartphones store and carry generated packets until their life-time (TTL) expires. If no Wi-Fi coverage is detected before the packet expires, data is immediately delivered through the cellular link and the transmission is assumed to be always successful.

Physical (PHY) and Medium Access Control (MAC) parameters of smartphones and APs are set to fit the IEEE 802.11g technology settings. Smartphones' transmission power P_{tx} and receiver sensitivity R_s values are set like in common devices on the market. Specifically, the smartphones in the simulated 100 vehicles belong to two classes, labelled as *type 1* and *type 2*; the former has higher P_{tx} and worse R_s than the latter.

Channel propagation is modeled through the Nakagami distribution, with fading parameter m set to 1 in order to simulate very harsh propagation conditions (i.e., Rayleigh fading).⁶ Simulation re-

sults are averaged over fifteen 20 minutes-long independent runs and reported with the 95% confidence interval.

7.2. Performance results

The performance of the offloading technique is evaluated in terms of: (i) *offloading percentage*, which is the percentage of augmented FCD packets that are transmitted through the Wi-Fi network rather than over the cellular links; (ii) the positive effect of offloading a single LTE cell in terms of saved radio resources (i.e., *resource blocks usage*); (iii) *average delay of offloaded packets*, computed as the time elapsed from the packet generation to its successful delivery to a Wi-Fi AP.⁷

⁶ It should be noticed that additional attenuation may be experienced in the vehicle due to the car body, as preliminarily investigated in [47]. These effects are not investigated in this work, but they will be a subject matter of future work.

⁷ Packets transmitted over the cellular links always experience the maximum delay, nearly equal to the TTL, since they are queued in the hope of detecting a Wi-Fi AP.



Fig. 12. Snapshot of the simulated map of the city of Rome: city center (a), not-far from city center (b), suburban area (c).

Fig. 14 reports the offloading percentage in the three zones and for the two considered different smartphones models when varying the TTL of augmented FCD packets. The longer the packet's lifetime, the higher the percentage of packets that go through the Wi-Fi hotspots. Indeed, a packet with a longer TTL can be stored for a longer time in the smartphone before being transmitted; hence the vehicle has more chances to meet an AP before the packet lifetime expires.

As expected, results clearly show that the offloading performance improves with the density of APs in the map. This is especially true when passing from 11 to 55 APs (the gain is between 20 and 30%), while less remarkable differences are noticed when passing from 55 to 97 APs (the gain is between 5 and 10%). Indeed,

the fact that in the city center the number of APs is nearly doubled with respect to an area of similar size farther from the center does not likewise imply an offloading performance improvement. This is due to the fact that APs typically are not regularly deployed in urban environments. For instance, a redundant overlapping coverage of APs may be available in some zones (e.g., close to points-of-interests), whereas other spots are not covered at all. The average number of detected APs at every second is reported in Table 6; results confirm that in the city center this number is higher than in the other zones. In general, the smartphone type 2, which has better receiver sensitivity, is able to successfully detect a higher number of APs.



Fig. 13. Deployment of APs in the map of the city center of Rome as from the website [45] (97 in the city center, 55 not far from city center, and 11 in the suburban area).

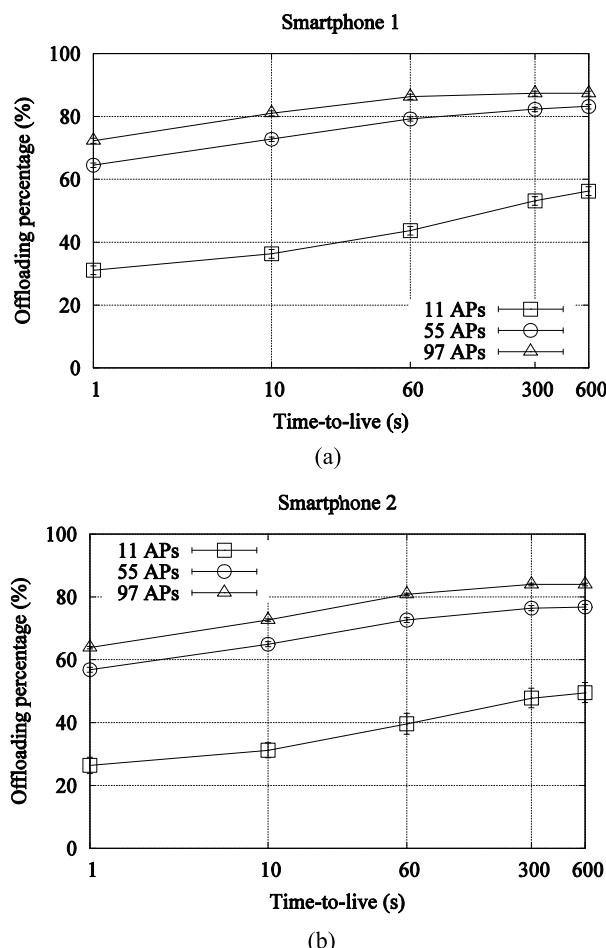


Fig. 14. Offloading percentage in the three zones vs. the time-to-live of augmented FCD packets (a) smartphone type 1; (b) smartphone type 2.

When comparing Figs. 14(a) and 9(b), it can be observed that better offloading performance is achieved by the smartphone with a higher transmission power (type 1). It has a larger transmission range, hence it is able to reach the AP from farther and offload a higher percentage of packets (nearly 10% more). However, even under the worst simulated conditions (i.e., type 2 smartphone, $TTL = 1$ s and 11 APs), a not negligible amount of packets can be offloaded, which is nearly one over four generated packets.

With the purpose of measuring more precisely the offloaded resources in the cellular network, we focus on an LTE single-cell scenario and compute the saving in terms of resource blocks (RBs) achieved when 11 APs are deployed in the cell area. We consider standard LTE specifications, as reported in Table 5. Frequency Division Duplexing (FDD) mode is assumed with 25 uplink RBs available in the cell. Simulations have been conducted when varying the number of vehicles (50, 75, 100, 125, 150) in the cell.

The RBs usage is reported in Fig. 15. It can be observed that the augmented FCD traffic asks between 6% and 15% of the cellular resources when no offloading is executed. It reaches the 15% when 150 vehicles transmit in the cell; this is a significant amount of uplink resources, subtracted to other applications supported by mobile network operators (mainly voice, web data traffic and emerging M2M communications).

If offloading through Wi-Fi is enabled, then the burden on the cellular network is significantly reduced. For instance, if 11 APs are available, then the achieved 25% and 50% offloading figures (with TTL set to 1 s and 300 s, respectively) turn out into an RB usage of 6.4% and 4.8% when 100 transmitting vehicles are considered. Saving resources is particularly beneficial to ensure that the capacity demands of other applications can be met.

The packet delay metric for offloaded packets is reported in Fig. 16. The delay increases with the lifetime of packets, though it is significantly lower than the TTL. The lower the number of APs the higher the suffered delay, since packets are queued for a longer time while waiting for an AP detection. Indeed, the queuing delay is the main contribution to the overall delay. The differences between the two typologies of smartphones are negligible, with a slightly lower performance for the smartphone of type 2 due to the lower transmission power.

Curves in Figs. 14–16 accounted for a set-up delay of 4 s. The offloading performance is also affected by the time required to set-up a connection with an AP (scanning, authentication, and association procedures), as illustrated in Fig. 17.

The longer the set-up time the lower the offloading percentage. This may be critical for short-lived packets (the shortest TTL values are reported in the curves), since they may expire before the association with the AP is completed after it has been detected.

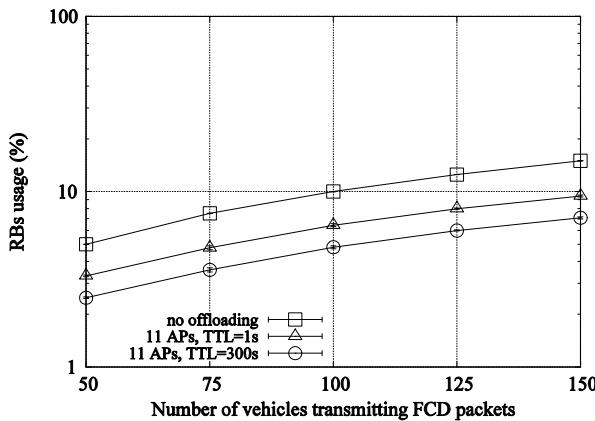
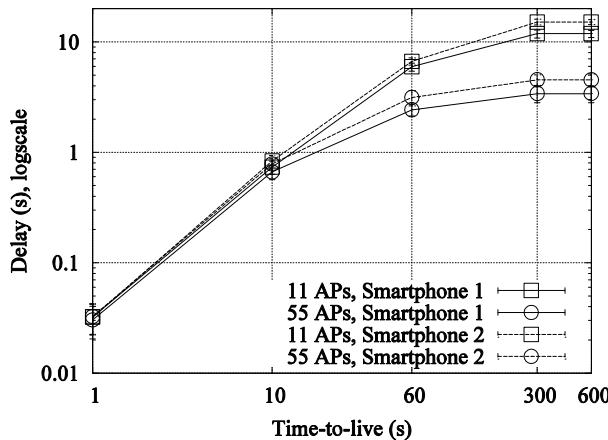
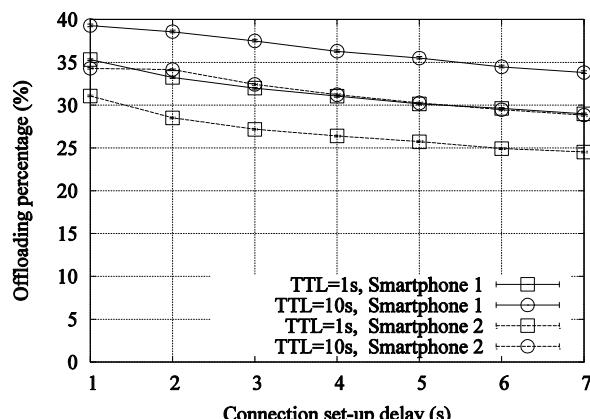
8. Conclusion

In this paper we discussed challenges and opportunities relevant to the support of day-one ITS applications based on augmented floating car data transfer.

Table 6

Average number of APs detected by different smartphones in each zone (sampled at every second).

	Suburban area (11 APs)	Close to city center (55 APs)	City center (97 APs)
Type 1 ($P_{tx} = 16 \text{ dBm}$, $R_s = -66 \text{ dBm}$)	2.133 (± 0.34)	10.012 (± 0.87)	17.491 (± 0.74)
Type 2 ($P_{tx} = 13 \text{ dBm}$, $R_s = -74 \text{ dBm}$)	3.039 (± 0.17)	14.882 (± 0.93)	27.264 (± 0.86)

**Fig. 15.** RB usage in an LTE network when varying the number of vehicles transmitting FCD packets and under different offloading conditions.**Fig. 16.** Delay of offloaded packets vs. time-to-live for different AP densities and smartphone typologies.**Fig. 17.** Offloading percentage vs. the Wi-Fi connection set-up delay under different time-to-live values and smartphone typologies (map with 11 APs).

After a careful analysis of the potential of FCD services and the raised issues for their large-scale adoption, we presented *Smart-Car*, our platform leveraging popular consumer electronics devices like smartphones and available networking technologies (cellular and Wi-Fi) for the support of FCD and urban sensing applications. Thanks to its short-time-to-market, the proposed solution can be used in place of sophisticated specialized IEEE 802.11p on-board communication devices for which the rulemaking in newly sold cars is still underway.

The proposed smartphone-based platform has the virtues of low-cost and simplicity; it can be adopted on a large-scale (i) by citizens and commuters, willing to contribute to data collection while benefiting from value-added on-the-road services (e.g., access to real-time traffic information, alternative route suggestions, connection to dealerships for service tips, etc.), and (ii) by fleet of vehicles to improve the customer satisfaction and their service efficiency.

The built prototype confirms the technological feasibility of the conceived platform. Evaluation conducted under a wide range of realistic simulation settings proves the effectiveness of offloading the cellular network by exploiting Wi-Fi hotspots coverage. Results show that the offloading strategy is highly sensitive to the lifetime of collected packets, and its benefits strongly depend on the delivery requirements of the application consuming data (real time or not). Moreover, the smartphone's Wi-Fi capability, the APs density, and the Wi-Fi connection set-up delay affect the offloading performance. In any case, the beneficial effect of offloading ranges from nearly 25%, in the worst analyzed scenario (TTL equal to 1 s, 11 APs), to more than 85%, for long-lived packets transmitted by smartphones with high power in a city center with high density of Wi-Fi hotspots.

To increase the AP coverage area, with additional benefits in terms of offloaded traffic, smartphones in other vehicles can be used to relay data towards the AP, as also debated in [23]. Relaying data through other vehicles is expected to increase the design complexity; it entails challenges in the relay selection (e.g., the vehicle geographically ahead or behind when approaching or leaving the AP coverage area, respectively). Moreover, the interference may get stronger; for example, when vehicles close to the AP also need to upload data, they compete for bandwidth with relay nodes, hence requiring proper scheduling schemes. In our designed solution, we kept simplicity as one of the main value-added features. However, relaying through other vehicles can be an extremely interesting opportunity to investigate for future work.

Further efforts are required also to design more sophisticated delivery strategies encompassing smart packet aggregation according to the semantics of collected data (e.g., degree of novelty and correlation patterns). Finally, attractive business models are a need to incentivize the user participation in data collection. Specifically, business models should foresee cooperation and agreements between the involved actors (end-users, Telco operators, and ITS service providers) to support the cost of the platform and incentivize commuters in taking part to mobile *crowd-sensing*. For instance, commuters could be rewarded with free access to some contents of interest, on the basis of the collected and transferred data volume, or with customized tips and diagnostics for their vehicles.

References

- [1] S. Bessler, T. Paulin, Literature study on the state of the art of probe data systems in Europe, September 10, 2013.
- [2] G. Pau, R. Tse, Challenges and opportunities in immersive vehicular sensing: lessons from urban deployments, *Signal Process. Image Commun.* 27 (8) (2012) 900–908.
- [3] 3GPP TR 37.868 V11.0.0, 3rd generation partnership project; technical specification group radio access network; study on RAN improvements for machine-type communications (Release 11), 2011-09.
- [4] ETSI TR, 102 898 v. 1.1.1, Machine-to-machine communications (M2M): use cases of automotive applications in M2M capable networks, 2013-04.
- [5] SAE, SAE J2735 dedicated short range communications message set dictionary, Technical report, SAE, 2007, <http://www.sae.org>.
- [6] C. Campolo, A. Molinaro, Multichannel communications in vehicular ad hoc networks: a survey, *IEEE Commun. Mag.* 51 (5) (2013) 158–169.
- [7] ACM VANET, Panel: Smartphone as vehicular networks data hub, <http://www.uwicore.umh.es/vanet2013/program.html>, 2013.
- [8] F. Gil-Castilleira, D. Chaves-Diéz, F.J. González-Castaño, Integration of nomadic devices with automotive user interfaces, *IEEE Trans. Consum. Electron.* 55 (1) (Feb. 2009) 34–41.
- [9] <http://openxplatform.com/>.
- [10] <https://www.onstar.com/>.
- [11] M. Fogue, P. Garrido, F.J. Martínez, J.C. Cano, C.T. Calafate, P. Manzoni, Assistance through communication technologies and vehicles, automatic accident detection, *IEEE Veh. Technol. Mag.* 7 (3) (September 2012) 90–100.
- [12] J. Zaldivar, C.T. Calafate, J.C. Cano, P. Manzoni, Providing accident detection in vehicular networks through OBD-II devices and Android-based smartphones, in: 5th IEEE Workshop on User Mobility and Vehicular Networks, Bonn, Germany, 4–7 October, 2011.
- [13] R. Frank, H. Weitz, G. Castignani, T. Engel, Collaborative traffic sensing: a case study of a mobile phone based traffic management system, in: Proceedings of the 11th Consumer Communications and Networking Conference (CCNC), January 2014.
- [14] K. Ali, D. Al-Yaseen, A. Ejaz, T. Javed, H.S. Hassanein, CrowdITS: Crowdsourcing in intelligent transportation systems, in: IEEE Wireless Communications and Networking Conference (WCNC), 2012.
- [15] J. Meseguer, C.T. Calafate, J.C. Cano, P. Manzoni, DrivingStyles: a smartphone application to assess driver behavior, in: Proc. – IEEE Symp. Comput. Commun., 2013.
- [16] R. Araujo, A. Igreja, R. de Castro, R.E. Araujo, Driving coach: a smartphone application to evaluate driving efficient patterns, in: IEEE Intelligent Vehicles Symposium, 2012.
- [17] H. Zhao, H. Zhou, C. Chen, J. Chen, Join Driving: a smartphone-based driving behavior evaluation system, in: IEEE Globecom, 2013.
- [18] <http://www.openautoalliance.net/>.
- [19] G. Remy, S.M. Senouci, F. Jan, Y. Gourhant, LTE4V2X-Collection, dissemination and multi-hop forwarding, in: IEEE ICC, June 2012.
- [20] R. Stanica, M. Fiore, F. Malandrino, Offloading floating car data, in: IEEE WoWMoM, 2013.
- [21] C. Ide, B. Dusza, M. Putzke, C. Wietfield, Channel sensitive transmission scheme for V2I-based floating car data collection via LTE, in: IEEE ICC, June 2012.
- [22] K. Lee, I. Rhee, J. lee, S. Chong, Y. Yi, Mobile data offloading: how much can WiFi deliver? in: ACM CoNEXT, 2010.
- [23] N. Cheng, N. Lu, N. Zhang, X. Shen, J.K. Mark, Vehicular WiFi offloading: challenges and solutions, *Veh. Commun.* 1 (1) (January 2014) 13–21.
- [24] C. Campolo, A. Iera, A. Molinaro, S.Y. Paratore, G. Ruggeri, SMaRTCaR: an integrated smartphone-based platform to support traffic management applications, in: First IEEE International Workshop on Vehicular Traffic Management for Smart Cities (VTM), 2012.
- [25] O. Briante, C. Campolo, A. Iera, A. Molinaro, S.Y. Paratore, G. Ruggeri, M. Booyesen, ItsPhone: an integrated platform for participatory ITS data collection and opportunistic transfer, in: IEEE INFOCOM 2013, Demo session, Turin, Italy, April, 2013.
- [26] <http://www.arduino.cc/>.
- [27] <http://www.android.com/>.
- [28] W. Huber, M. Lädke, R. Ogger, Extended floating-car data for the acquisition of traffic information, in: Proceedings of the 6th World Congress on Intelligent Transport Systems, 1999.
- [29] C. Ide, B. Dusza, C. Wietfeld, Energy efficient LTE-based floating car data collection for dynamic traffic forecasts, in: IEEE International Conference on Communications (ICC), 2012.
- [30] D.F. Llorca, Traffic data collection for floating car data enhancement in V2I networks, *EURASIP J. Adv. Signal Process.* 5 (2010) 1–13.
- [31] Cohda Wireless, CohdaMobility MK2 WAVE-DSRC radio, available at <http://www.cohdawireless.com/product/mk2.html>.
- [32] Unex 5.9 GHz DSRC channels, available at <http://unex.com.tw/dsrc-wave/dsrc-mini-pci>.
- [33] ETSI TS 102 637-3, Intelligent Transport Systems (ITS); Vehicular communications; Basic set of applications; Part 3: Specifications of decentralized environmental notification basic service, 2010.
- [34] <http://topics.sae.org/obd/standards>.
- [35] <http://developer.android.com/tools/adk/index.html>.
- [36] A.S. Zeeman, M.J. Booyesen, G. Ruggeri, B. Laganá, Capacitive seat sensors for multiple occupancy detection using a low-cost setup, in: IEEE International Conference on Industrial Technology (ICIT), 2013.
- [37] <http://arduino.cc/en/Main/ArduinoXbeeShield>.
- [38] S. Rass, S. Fuchs, M. Schaffer, K. Kyamakya, How to protect privacy in floating car data systems, in: Proc. of ACM VANET, 2008.
- [39] R. Nikodin, J.Y. Le, Boudec, A. Chaintreau, V. Erramilli, Energy efficient offloading of 3G networks, in: IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS), 2011.
- [40] S. Nawaz, C. Efstratiou, C. Mascolo, ParkSense: a smartphone based sensing system for on-street parking, in: ACM International Conference on Mobile Computing & Networking (MobiCom '13), 2013.
- [41] S. Devarakonda, P. Sevusu, H. Liu, L. Iftode, B. Nath, Real-time air quality monitoring through mobile sensing in metropolitan areas, in: Proc. of the 2nd ACM SIGKDD International Workshop on Urban Computing, 2013.
- [42] Network Simulator, Ns-2.
- [43] OpenStreetMap, www.openstreetmap.org/.
- [44] SUMO, Simulation of Urban Mobility, <http://sumo-sim.org/>.
- [45] <http://www.mappawifi.provincia.roma.it>.
- [46] W. Haitao, K. Tan, J. Liu, Y. Zhang, Footprint: cellular assisted Wi-Fi AP discovery on mobile phones for energy saving, in: Proc. of the 4th ACM International Workshop on Experimental Evaluation and Characterization, 2009.
- [47] M. Blesinger, et al., Angle-dependent path loss measurements impacted by car body attenuation in 2.45 GHz ISM band, in: IEEE Vehicular Technology Conference, VTC, Spring 2012, 2012.