# A Novel On-Demand Vehicular Sensing Framework for Traffic Condition Monitoring

Sawsan Abdul Rahman[a], Azzam Mourad[a,*], May El Barachi[b], Wael Al Orabi[a]

[a]*Department of Computer Science & Mathematics, Lebanese American University, Beirut, Lebanon*
[b]*Department of Engineering & Information Sciences, University of Wollongong Dubai, Dubai, U.A.E.*

**Abstract**

With the increased need for mobility and the overcrowding of cities, the area of Intelligent Transportation aims at improving the efficiency, safety, and productivity of transportation systems by relying on communication and sensing technologies. One of the main challenges faced in Intelligent Transportation Systems (ITS) pertains to the real time collection of traffic and road related data, in a cost effective, efficient, and scalable manner. The current approaches still suffer from problems related to the mobile devices energy consumption and overhead in terms of communications and processing. To tackle the aforementioned challenges, we propose in this paper a novel infrastructure-less on-demand vehicular sensing framework that provides accurate road condition monitoring, while reducing the number of participating vehicles, energy consumption, and communication overhead. Our approach is adopting the concept of Mobile Sensing as a Service (MSaaS), in which mobile owners participate in the data collection activities and decide to offer the sensing capabilities of their phones as services to other users. Unlike existing approaches that rely on opportunistic continuous sensing from all available cars, this ability to offer sensory data to consumers on demand can bring significant benefits to ITS and can constitute an efficient and flexible solution to the problem of real-time traffic/road data collection. A combination of prototyping and traffic simulation traces are used to realize the system, and a variety of test cases are used to evaluate its performance. When compared to the traditional continuous sensing, our proposed on-demand sensing approach provides comparable high traffic estimation accuracy while significantly reducing the resource consumption. Based on the obtained results, using the on-demand sensing approach with 30% of cars as participants in the sensing activity, and a six-criteria matching approach yields a reduction of 73.8% in terms of network load and a reduction of 60.3% in terms of response time (when compared to the continuous sensing approach), while achieving a traffic estimation accuracy of 81.71%.

*Keywords:* Sensing as a Service; Intelligent Transportation Systems; Traffic Estimation; On Demand Sensing; Road Condition Monitoring.

## 1. Introduction

With the rapid widespread of smartphones that come embedded with a variety of sensors (e.g. gyroscope, GPS, and accelerometer), users now hold in the palms of their hands powerful devices that can be used as personal sensing platforms enabling the collection of a wealth of contextual information. This integration of sensing technology in mobile devices opens the door for a new sensing approach and era [1]. Mobile devices can act as super sensors that are readily deployed and can be used to dynamically collect intelligence about cities. There are two main mobile phone sensing paradigms: Participatory sensing in which the user actively participates in the data collection and sensing activity; and opportunistic sensing that occurs in a transparent automated manner without any user involvement [1]. Furthermore, different sensing modes can be adopted, namely: Sense-once, Event-based sensing, Time-based sensing with expiry duration, and continuous sensing.

Sensing technologies constitute one of the key enablers of Intelligent Transportation Systems (ITS). In fact, ITS rely on communication and sensory technologies along with data processing and analysis techniques to improve the safety, efficiency, and productivity of transportation systems [2]. Typical ITS applications include traffic management, road safety applications, and route planning applications. The collection of real time traffic and road conditions constitutes an important challenge in such applications. Conventional methods for the collection of such information typically relied on infrastructure sensors such as surveillance cameras and inductive loops, which may not be always available and involve high deployment and maintenance costs [3]. Recently, the idea of using mobile crowdsensing for the collection of traffic and road related information [4] has attracted attention in academic and industrial forums. In this approach, regular users equipped with sensor-enabled phones collaborate to sense data related to phenomena of interest (e.g. traffic conditions and accidents' occurrence) [5]. The reliance on the drivers carrying sensor-embedded phones for the collection of traffic related information brings important benefits. The first benefit pertains to the easy on-demand deployment of a large-scale network of sensors, since millions of mobile phones are carried everyday by

*Corresponding author.
*Email addresses:* sawsan.abdulrahman@lau.edu (Sawsan Abdul Rahman), azzam.mourad@lau.edu.lb (Azzam Mourad), maielbarachi@uowdubai.ac.ae (May El Barachi), orabiwael15@gmail.com (Wael Al Orabi)

vehicle drivers. Moreover, this approach leads to important time saving and costs reduction with respect to traditionally deployed specialized sensing infrastructures. Examples of mobile crowdsensing systems used in the area of intelligent transportations include MIT's CarTel [6] and Microsoft Research's Nericell [7]. These systems mainly adopt a continuous sensing approach in which data is continuously sampled from all cars on all street segments (without the explicit involvement of users), and then processed offline on the back-end server. However, this imposes high energy-requirements on mobile devices, entails significant overhead on the mobile communication infrastructure, and results in large amounts of data requiring processing on the server. Furthermore, the opportunistic automated data collection strategy adopted by such systems gives rise to privacy concerns by mobile users, which may not wish to share sensory data that reveals sensitive information about themselves (e.g. their geographic location).

Moreover, in similar context, the connected vehicles technology [8] has emerged recently, which enables the communication between vehicles (i.e. Vehicle to vehicle) as well as between vehicles and the roads' infrastructure (i.e. vehicle to infrastructure), using dedicated short range communications (DSRC) [9]. **In [10], a traffic estimation for highway was estimated using connected cars with and without being equipped with Adaptive Cruise Control.** However, despite the merits of the connected vehicles technology [11] and its potential use for safety and congestion management applications, this technology presents certain limitations when compared to the mobile crowdsensing technology. The first limitation pertains to the smaller market penetration rate of connected vehicles (fore casted to reach 152 million connected vehicles sold by 2020 [12]), when compared to the massive and pervasive market penetration of smart-phones that have passed the 2 billion device mark in 2016 and are expected to reach 2.87 billion in 2020 [13]. In fact, the effectiveness of ITS relying on sensory technology depends on the sufficient penetration of the technology in streets, a fact that cannot be currently guaranteed with connected vehicles, but can be easily achieved with smartphones. Furthermore, smart vehicles [14] currently face limitations in terms of their communication and sensing capabilities, under adverse weather conditions.

Recently, the Mobile Sensing as a Service (MSaaS) approach has been emerged [15, 16], in which mobile devices and users willingly participate in the sensing process and offer their phones' sensory data collection capabilities as services to other users. This approach is very promising to address the aforementioned issues, and to the best of our knowledge, none of the previous related works consider it in ITS solutions. In this work, we propose a novel vehicular sensing framework enabling on-demand road condition monitoring in efficient and flexible manner. Unlike existing solutions that rely on opportunistic continuous sensing from all cars available, we advocate participatory on-demand sensing from a selected number of cars that can offer a high quality of sensed information. In the proposed framework, status and traffic data sensed about any region of interest would occur on demand, when triggered by a sensing request. Once the sensing request is received by the sensing platform from a data consumer, the set of targeted users acting as data collectors will be determined by the platform based on a proposed multi-criteria matching algorithm that takes into account the collectors' presence in the region of interest, their phones' sensing capabilities, the users' willingness to participate in the sensing activity, the users' reputation, the phones' battery level, and the accuracy of the data they provide. Once the sensed data is received from the targeted data collectors, the sensing platform relies on a traffic estimation algorithm to estimate the traffic condition, which is sent in the form of a traffic report to the user who sent the original sensing trigger request. More interesting scenarios could be enabled by the concept of on-demand vehicular sensing as a service such as "On-Demand Accident Scene Intelligence Gathering" and "On-Demand Road Condition Monitoring". Accident scene such as stationary cars, injuries and Road status such as traffic level, snow removal conditions, potholes in streets, fog or bad weather conditions, road redirection can be collected on demand by the sensing platform, analyzed and then provided as a report to the user triggering the sensing request. It should be noted that the messages exchanged between the platform and the users (acting as data collectors and data consumers) is conducted using RESTful web services communication interfaces which we defined for our framework.

In order to study the performance of our vehicular platform and compare the on-demand sensing approach to the traditional continuous sensing approach, we combined prototyping and simulated traffic traces to build a proof-of-concept prototype of the system. Furthermore, we conducted extensive experiments in which different parameters were varied, such as: the traffic conditions on the road, the matching criteria used for participants' selection, the number of sensing requests received by the platform/hour, the frequency of voluntary data publication requests, and the percentage of cars participating in the sensing activities. Four main performance metrics were measured using various test cases, namely: The traffic estimation accuracy, the participants' selection accuracy, the system's response time, and the system's network load. This comparative performance analysis gives interesting insights on the contributions and benefits of an-demand participatory sensing approach, and the trade-offs that can be achieved between the data collection frequency, the percentage of cars participating in the data collection activity, the traffic estimation accuracy, and the system's performance.

In summary, the main contributions of our participatory on-demand sensing framework are five folds:

- **High Traffic Estimation Accuracy**: Our proposed approach is able to successfully infer the traffic status in all the tested scenarios. The experimental results show that the estimation error % decreases from 17.82% for 10 sensing requests received/hour to 2.9% for 10000 sensing requests received/hour in the on-demand approach. Such results are comparable to the standard continuous approach, in which the estimation error decreases from 17.7% for data voluntarily published each 10 minutes to 5.9% for data published each 30 seconds.

- **Reduced Resource Consumption**: Due to the fact that the number of cars involved in on-demand sensing is reduced to a selected set of cars present in the area of interest, chosen based on several selection criteria, and approached on need basis only (instead of continuously publishing their information), important reductions in the amount of generated network load, energy consumption on mobile devices, and amount of data requiring processing on the server can be achieved. Moreover, our proposed approach strikes a balance between traffic estimation accuracy and resource consumption. This is achieved by using contextual information and a multi-criteria participants' selection approach to select the smallest number of data collectors that can provide the best quality of sensed data, in order to maintain a good traffic estimation accuracy and an improved system performance (i.e. lower response time and network load).

- **High Quality of Sensed Information**: Our approach relies on a multi-criteria selection approach that enables the achievement of a high Quality of sensed information, by selecting the best candidates yielding the highest quality records satisfying multiple quality of information criteria. This participants' selection approach leads to more accurate traffic estimation results.

- **Users' control over their devices related information**: Our participatory sensing approach offers more control to mobile phone users over the sensed data collected using their devices, since users can accept or deny a sensing request. This is not the case in opportunistic continuous sensing which is typically performed systematically, without the involvement/consultation of users.

- **Flexible and Individual Sensing as a Service Operations**: Our approach allows more flexibility and availability of data in any area of interest through direct individual agreements with the user (i.e. no need for agreements with transportation and telecommunication authorities), by tapping into the sensing capabilities of millions of mobile phones deployed across the glob to obtain traffic conditions on any street of interest.

The rest of the paper is organized as follows: Section 2 discusses the related work. Section 3 details the proposed vehicular sensing framework. Section 4 is dedicated to the description of the proposed traffic estimation and participants' selection models. This is followed by the prototype and implementation in section 5 and the experimental results in Section 6. We end the paper with our conclusions, in Section 7.

## 2. Related Work and Problem Statement

Several works on sensing for the continuous collection of traffic and road related information have been recently carried out. In this approach, a group of users having sensor-enabled devices (e.g. mobile phones, GPS readers) collectively sense relevant data to estimate the traffic condition in a specific area of interest. Moreover, there is a rich literature providing traffic estimation approaches that rely on specialized sensing devices embedded into Intelligent transport infrastructure. In the following, we elaborate the main related approaches in addition to the technical problem statement at the end of the section.

In [17], the authors proposed the use of GPS and accelerometer data for the detection of traffic conditions, abnormalities, and potholes on roads. This approach consists of five components: smartphones, a local database (for temporary storage of data), open wireless networks, a server hosting a central database, and open street maps. The sensed data is sent to a heuristic algorithm that analyzes it and produces roads' traffic status. Herring et al. [18] proposed a solution that targets traffic conditions on highways. The model consists of one physical component which is the GPS, and three cyber components: a cellular network operator, cellular phone data aggregation and traffic service provision, and traffic estimation algorithms. In this approach, data is collected using mobile phones on specific trajectories called virtual trip lines. This data is sent to a server that aggregates it and sends it to the Ensemble Kalman Filtering based traffic estimation algorithm. In [19], Thiagarajan et al. proposed an approach to overcome energy consumption and inaccurate position sampling challenges by using a Hidden Markov Model (HMM) that depicts the trajectory of a vehicle over a portion area in the map. They performed map matching in order to estimate the travel times of the traversed road segments. In [7], Mohan et al. proposed a solution called NeriCell that focuses on the sensing component such as accelerometer, microphone, GSM radio, and GPS sensors. They used Intelligent Traffic System that needs dedicated sensors in streets and cars. This solution consists of a system of rich monitoring of road and traffic conditions that piggybacks on smartphones and calculates roads' traffic status using vehicles' acceleration data. Herreraa et al. proposed two data gathering techniques (spatial and temporal) in [20]. Spatial sampling implies that equipped vehicles report their information (position, velocity, etc..) at specific time intervals T regardless of their positions, while temporal sampling implies that the vehicles report their information as they cross some spatially defined sampling points. In this approach, data is collected from mobile devices (Nokia N95) every 3 seconds, then the instantaneous velocity is measured at the same rate, and these data will form a rich history of data used for traffic estimation. Also, they targeted and solved the privacy aspect concerning the identity of the users. Recently, the authors in [21] proposed a distributed peer-to-peer approach to traffic estimation. In this approach, a car uses V2V communication to collect position and velocity related data from nearby cars. The data collected is sparse data in the form of floating car data snapshots and the Underwood traffic-engineering model based on density is used for traffic condition estimation.

In another context, the following proposed approaches were focusing on the importance of sensing as a service by mobile phone sensors: Ban and Gruteser, in [22], focused on two important issues. The first one is fine-grained urban traffic knowledge extraction, while the second is the privacy protection scheme. They provided a comparison between the primary way of collecting through fixed-location sensors and the newly suggested one through the mobile phones sensors. Based on their claims, collecting data through fixed-location sensors costs a lot and is not an efficient way in order to predict traffic efficiently, while collecting and extracting data through mobile phones will greatly benefit the urban traffic prediction applications in terms of performance. This type of collecting data can provide detailed behaviors and continuous trajectories of the vehicles. In [23], Khan et al. conducted a survey that talks about the different monitors and usages of mobile sensing which are: health, traffic, environment, social, special purpose, human behavior, and commerce. It mainly distinguishes between two types of urban sensing. The first type is the participatory, while the second is the opportunistic. In both types of sensing, the solutions implemented are divided into three main parts: personal, public, and social. In each of the solutions, authors emphasize the used type of sensors, hardware and software description, communication modules, and applications. In [24], Das et al. did not target the traffic estimation on roads problem, rather they focused on the community sensing (participatory and opportunistic). They focused on the community sensing which targets the embedded sensors on the mobile phones such as GPS, camera, audio, accelerometer, and GSM. The main goals of their paper are to ensure (1) generality by supporting a wide range of applications with flexibility of reusing existing code, (2) security by ensuring that the participating phones belonging to individual users remain secure and that the applications do not misuse sensitive sensor information, and (3) scalability by allowing the system to scale to a large number of nodes without placing an undue burden on the infrastructure itself. In [25], Placzek focused on the idea of reducing the amount of data transmitted through Vehicular Sensor Network in order to control the roads traffic. Instead of periodically requesting sensed data from vehicles, the proposed approach specifies time moments when the queries should be sent. The selected time moments are characterized by the uncertainty of traffic estimation, and in this case, new traffic data is requested.

**Problem Statement**: All the aforementioned mobile sensing related approaches rely on continuous or periodic sensing of road and traffic data, which entails the following problems:

- High energy consumption on mobile devices due to the continuous sensing from the relevant sensor such as GPS, accelerometer, etc.

- Communication overhead on mobile infrastructure due to (1) continuous data sensing from each vehicle and (2) data collection from all the vehicles without any filtering/selection criteria during collection. All the customizations performed by these approaches are done at the server side, i.e. during traffic analysis after collection. The impact of this problem will potentially increase with the fast emergence of Internet of things that will overhead the mobile infrastructure, reaching around 29 Billion connected devices by year 2022 [26].

- High processing overhead and full availability of road data since the current traffic analysis models and algorithms are dependent on continuous and complete data collection from all vehicles in order to estimate the mean speed, density, and flow.

In order to address the aforementioned problems, the proposed framework offers on-demand and upon need data collection gathered based on several selection criteria such as availability, location, need, etc. To the best of our knowledge, none of the current approaches in the literature have targeted the aforementioned problems and addressed on-demand sensing in the context of ITS and traffic estimation.

## 3. Vehicular Sensing Framework Overview

Figure 1 depicts the high-level architecture of the proposed vehicular sensing framework. Our system encompasses three main roles: Data consumers interested in the acquisition of sensed data related to a particular area of interest within the city (e.g. provide me with traffic conditions or snow clearance conditions on road X); data collectors offering their phones' data collection/sensing capabilities as services to other users; and vehicular sensing platform acting as intermediary and data broker between consumers and collectors. The vehicular sensing platform receives sensing requests from data consumers and matches those requests with the most suitable data collectors based on some selection criteria. Afterwards, the platform sends the sensing request to the chosen data collectors through the matching model, who can either accept or reject it. Those who accept the request would perform the required sensing task and send the sensed data to the vehicular sensing platform, which is responsible of validating, aggregating and processing it through the relevant model and algorithms, and then sending the reply to the requester. The communication between the different roles can occur either using mobile communication infrastructures (e.g. 3G/4G mobile networks) or over public WiFi hotspots if available (e.g. in smart cities).

### *3.1. Components Description*

We now describe the functions performed by our system's entities in more detail:

- **Data Consumer**: The data consumer is a user who is interested in sensing services. To access those services, the data consumer interacts with the vehicular sensing platform through a gateway application to discover the sensing communities available. Once subscribed to a sensing community, the data consumer can discover and subscribe to (all or some of) its associated services. An example of a sensing community could be "New York city drivers" and examples of sensing services are "Traffic condition monitoring service" and "Snow clearance notification service". After subscribing to sensing services, a data consumer can send a sensing trigger to the vehicular sensing platform by specifying the requested data type and sensing mode (i.e. sense once, event-based sensing, or continuous sensing), as well as the geographical area of interest.

- **Data Collector**: A data collector is a user equipped with a sensor-enabled mobile device, and who is willing to offer its data collection capabilities as services to other users. The mobile device should host a sensing gateway application enabling the interaction with the vehicular sensing platform. To offer sensing services, a data collector must first subscribe to become

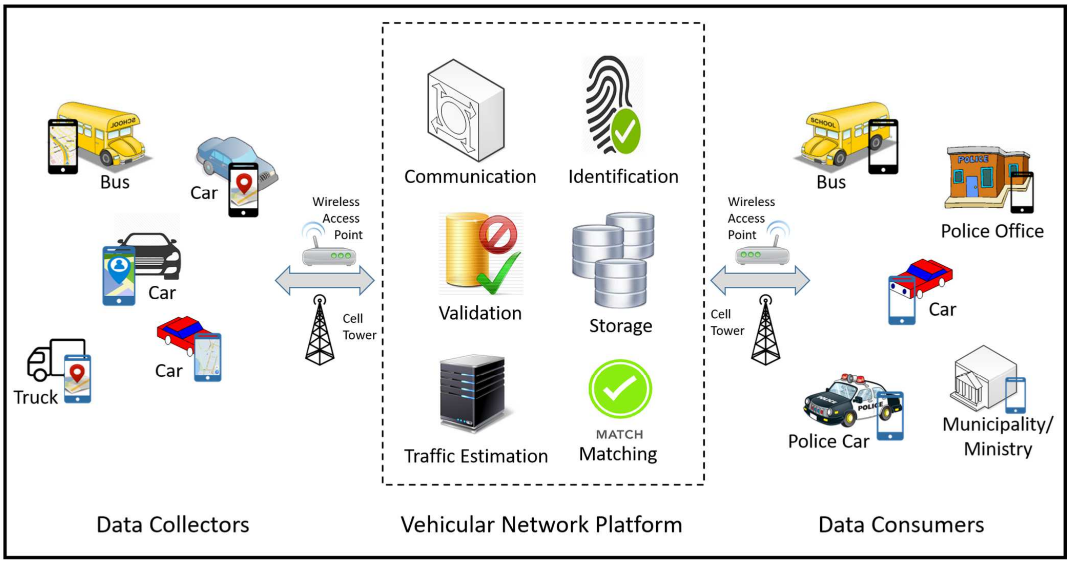



Figure 1: High Level Vehicular Sensing System Architecture

part of a sensing community. After subscription, the data collector periodically publishes his/her availability to the sensing platform (e.g. available, busy, and away) to indicate willingness to participate in sensing activities. The data collector's sensing gateway application should support a number of functionalities, including: handling sensing trigger requests from the platform; allowing the user to initiate sensing without trigger (i.e. offer-based sensing) and send the captured data to the sensing platform; ability to collect requested data from embedded sensors; supporting some information processing and formatting capabilities; providing Geo-temporal tagging of the sensed information; scheduling of sensing tasks; and management of sensing sessions based on received requests.

• **Vehicular Sensing Platform**: The vehicular sensing platform constitutes the key entity in our architecture. It acts as intermediary between data consumers and data collectors by matching sensing requests (in real time) with the most suitable data sources, and offers information management and data brokerage capabilities. To achieve that role, the vehicular sensing platform consists of a number of modules, namely: *communication, request handling, storage, validation, matching, identification, traffic estimation, analysis and reporting, and community membership modules.* The *communication module* is responsible of creating the communication messages (requests and responses) exchanged between the platform and the users. The *request handler* is responsible of identifying the type of message received and forwarding it to the appropriate module for further processing. The *storage module* is responsible for storing sensing activities related information. The *validation module* is responsible of the pre-processing of the collected information to detect inconsistencies and calibrate data. The *matching module* is a key module implementing a matching algorithm that relies on certain criteria (e.g. location and availability of data collector, data collection capabilities, data accuracy, available battery level, and user's reputation) to match sensing requests with the most suitable set of data collectors. The *identification module* is responsible of assigning unique IDs to the sensed entities, the sensing services offered, as well as users' roles in the system. The *traffic estimation module* processes the raw sensed data and produces traffic status information based on the proposed traffic estimation model and algorithm. The *analysis and reporting module* is used in some scenarios to generate advanced reports from collected data (e.g. accident scene summary reports). Finally, the *community membership module* keeps track of sensing communities, their related sensing services, as well as their subscribed users. We provide in the following sections the technical details of the models and algorithms deployed in the sensing platform.

### *3.2. Web Service based Communication Module*

The communication module in the sensing platform is responsible of handling the messages among the system entities. In the following, we present in details the type of the exchanged messages. In our vehicular sensing system, the communication between the components should be flexible and light weight since the platform supports multiple sensing requests and handles their response in parallel. Therefore, we select RESTful [27] as it is the best to work for mobile Web services and web-based applications. Representational State Transfer (REST) is an architectural style where data is considered as resources and accessed through Uniform Resource Identifiers (URIs).

Each entity in our system is thus considered as a web service that communicates through REST APIs. The exchanged messages, illustrated in Table 1, use the HTTP protocol and its most commonly used operations: POST, GET, PUT, and DELETE. POST creates a new resource and its URI will be automatically generated. GET reads the information about the resource in an appropriate representation. PUT updates the resource that can be deleted through DELETE. In the proposed model, our data set consists of Sensing Sessions split into four resources: "SensingSession", "DataConsumer", "DataCollector" and "Traffi-

| Resources | URI<br>Base URL:<br>http://VehicularSensing.com | HTTP<br>action/description |
|---|---|---|
| **Sensing Session** | /SensingSession | POST: create a new sensing session<br>GET: get all sessions |
| | /SensingSession /{SensingSessionID} | GET: retrieve a session. PUT: update a session. DELETE: terminate a sensing session |
| **Data Consumer** | /DataConsumers /{DataConsumerID} | GET: get info about a data consumer. PUT: update a data consumer info. DELETE: remove a data consumer from a session |
| **Data Collector** | /DataCollectors | POST: create a new data collector GET: get all data collectors |
| | /DataCollectors /{DataCollectorID} | GET: get info about a data collector. PUT: update data collector's status info. DELETE: delete a data collector |
| **Traffic Report** | /SensingSession /SensingSession ID/traffic | GET: get traffic report related to a sensing session PUT: update traffic report info. DELETE: remove a traffic and dissociate it from session |

Table 1: Web Services Communication Interfaces

cReport". Each SensingSession has one DataConsumer willing to get the traffic status of a particular road and a set of DataCollectors located on the specified road and a TrafficReport generated after processing the sensed data. The "SensingSession" resource is identified by the URI "http://VehicularSensing.com/SensingSessions/SensingSessionID" where SensingSessionID is the unique identifier of the Session. The "DataConsumer" is identified by "http://VehicularSensing.com/DataConsumers/DataConsumerID" where DataConsumerID is the identifier of the requester. The "DataCollector" is identified by "http://VehicularSensing.com/DataCollectors/DataCollector ID". The "TrafficReport" is identified by "http://Vehicular Sensing.com/SensingSessions/SensingSessionID/traffic". Table 1 summarizes also the URIs used in column 2, along with their operations in column 3 in order to access each resource found in the first column.

It should be noted that the data encompassed in the different HTTP messages and exchanged between the client and the server is formatted as XML documents. For instance, when a POST message is used to create a new sensing session, the data encompassed in this message sent from the client to the server is as follows:

<sensingSensing>
<dataType>traffic condition</dataType>
<sensingMode>sense once</sensingMode>
<areaofInterest>
<country>Canada </country>
<city>Montreal </city>
<street>Peel</street>
</areaofInterest>
<expiryTime>10-APR-17 15:30</expiryTime>
</sensingSession>

whereas the response sent back to the client contains a uniquely created session ID and identifier for the sensing session, such as:
"http://www.VehicularSensing.com/SS1357@CA-10042016"

## 4. Participants' Selection and Traffic Estimation Models

The key models in our Vehicular Sensing platform are the matching/participant selection model and the traffic estimation model. We focus in these models on the scenario where a data consumer sends a sensing trigger request to the sensing platform, with the following parameters: Data type = traffic condition; sensing mode = sense once; Area of interest = name of street on which sensing is required. Once the user sends the request, the server first runs the algorithm realizing the Matching model to retrieve the most suitable set of data collectors along with their sensed data. Then it runs the algorithm implementing the Traffic Estimation model to process the raw sensed data and predict the traffic status. All the notations of the used formulas in these two models are illustrated in Table 2.

| Variable | Description |
|---|---|
| $R$ | The desired road from which the traffic condition is inferred |
| $R'$ | An adjacent road heading toward $R$ |
| $R''$ | An adjacent road heading from $R$ |
| $pos_l^s$ | Last position of sensor $s$ at time $t_1$ |
| $pos_{cur}^s$ | Current position of $s$ at time $t_2$ |
| $s.avail$ | The availability of $s$ |
| $s.rep$ | The reputation of $s$ |
| $s.capab$ | The capability of $s$ |
| $s.dataAcc$ | The data accuracy of $s$ |

Table 2: Formulas Notations

### 4.1. *Matching and Participants' Selection Model*

The matching model is needed to retrieve the appropriate set of collectors whenever the platform's server receives a sensing request. In this context, several models have been advanced to select the suitable set. The participatory sensing framework proposed in [28] selects the social sensors and enables to share data based on their availability, trust and energy. To predict the user location and estimate his availability, an algorithm called Dynamic Tensor Analysis (DTA) is adopted since the user historical trajectory is known through his daily routine. All users with similar trajectories are clustered in 'Friends-Like Social Sensors' group where only one is selected to avoid the same data collection from multiple participants. The same concept has been proposed in [29] on how to choose the best set from a huge number of collectors and retrieve sensing data from them. The model focuses on finding not only the best set but also the minimum number of participants in the set that covers a given area of interest and satisfies certain constraints. The sensing requests can be sent at any time and handle both temporal and special requirements. The authors in [30] cover a certain area of interest based on the budget constraint by focusing on the scenario where the entire targeted region is divided into several sub-regions. The participants in each sub-region set specific prices in order to respond to the sensing requests and thus the system picks the ones with lowest prices to maximize the number of collectors. Some interpolation methods could be used in case the incentive budget is not sufficient or no collectors are located in the desired sub-region. In [31] and [32], the data consumer sends sensing tasks to the system server where several requirements are associated to the tasks such as the sensing area, time, data granularity and quantity. The proposed selection models in [31] and [32] allow to gather the maximum number of sensory collectors while minimizing the consumption of energy for all the participants. However, all of them did not target ITS and Traffic condition monitoring, which limit their relevance to our approach. Moreover, they are missing many important criteria needed for a traffic decision model.

In this context, we propose in the sequel a new matching model that considers several criteria for selecting the minimal set of sensing vehicles. The first criterion is the geographic location of the targeted collectors, which takes into consideration two cases; in the first one the data is collected from the cars located on the targeted road, while in the second where the data is collected from the cars that are heading toward the desired road and will eventually be located on it after a certain time frame. In order to find out the position of each participating collector without nullifying the on-demand sensing concept, each data collector must share periodically its recent sensed data with the server. Accordingly, the first case can be calculated since both the positions of the participating nodes and the road coordinates are predefined. However, concerning the second case, we determine a bounding circle around the middle of the targeted road and take the nearby streets that fall in this circle area based on the map topology. All cars located on the nearby streets are then added to the set of data collectors. The second criterion in the matching is the availability of the user. The status of the user is checked whether available or not to recognize if he is willing to participate in the sensing activities. At the time a user sends a non-availability, the server should not consider him in the set of collectors even if he is located in the desired area of interest. The third criterion is the battery level of the users' mobile phone. If the phone battery level of a user is less than or equals to 20%, then our matching approach assumes that the mobile phone is not capable of sending/receiving any form of data to/from the server and thus the user is not in the appropriate set of collectors. **The fourth criterion is the user's reputation, which helps to improve the performance of the platform. Every time a user participates in a data collection activity, we compare his response with the actual traffic condition to recognize if he misbehaved or not. Accordingly, users with bad reputation are excluded from further involvement in the traffic monitoring activities to not reduce the accuracy of our results. Moreover, users should be motivated to behave correctly [33, 34, 35] and guarantee their participation in the coming sensing activities with high reputation. Therefore, we made the bad performance of the user as data collector affects him negatively when taking the role of a consumer. When intending to request a traffic condition for his benefit, the user will be consequently prevented from any request.** The fifth criterion is the sensing capabilities of the user. This is useful in the general case where the user is sensing data related to temperature, CO2 level, or any other type of information. We need to check if the user is capable of sensing such type of data since not all the phones embed variety of sensors. The sixth and final criterion is the accuracy of the data sent by the user based on the type of the used sensor. For instance, the data collected using GPS is considered more accurate than the one collected using Wi-Fi.

In the sequel, we present the model combining all the aforementioned criteria followed by its corresponding algorithm (Algorithm 1). All the participating sensors S with total size n are first selected as input to find the initial $S_{init}$ defined by

$$\begin{aligned} S_{init} = \sum_{s=1}^{n} &(pos_l^s \in R \vee pos_l^s \in R') \quad \wedge \\ &(s.avail == true) \wedge (s.rep == high) \quad \wedge \\ &(s.capab == true) \wedge (s.dataAcc == good) \end{aligned} \tag{1}$$

where $S_{init}$ set holds all the sensors which their last position $pos_l^s$ at time $t_1$ was either on R or heading toward R, and are characterized by the following properties: are available, have high reputation, capable to sense the required data, and have good data accuracy. Let $t_1$ be the time when the participants shared the last sensed data with the server just before receiving a traffic request from a consumer and $t_2$ be the request time.

Since the server requires two recent sensed data for the collectors to estimate the roads conditions, the server sends sensing requests to each car in the list and gets their new positions and speeds as response. Hence, once the collectors in the set $S_{init}$ are found, the platform sends them sensed request to collect the appropriate data in order to estimate the road condition. The new set of sensors $S_{final}$ collected after receiving $S_{init}$' responses is defined by

$$\begin{aligned} S_{final} = \sum_{s=1}^{S_{init}.size} &(pos_l^s \in R \wedge pos_{cur}^s \in R) \quad \vee \\ &(pos_l^s \in R \wedge pos_{cur}^s \in R'') \quad \vee \\ &(pos_l^s \in R' \wedge pos_{cur}^s \in R) \end{aligned} \tag{2}$$

where set $S_{final}$ contains the cars that are located on R at time $t_2$ and were located on R' at time $t_1$, the cars that are located in the destination R at both $t_1$ and $t_2$, and the cars that were found on the road R at $t_1$ and becomes on its adjacent R" at $t_2$. Accordingly, the cars, heading toward the desired destination and changed their directions, are removed from the list and the rest will be sent to the traffic estimation module.

### *4.2. Traffic Estimation Model*

Once the vehicular platform successfully performs the matching process, the platform forwards the two sensed data $pos_l^s$ and $pos_{cur}^s$ for each sensor s in the set of collectors $S_{final}$ to the traffic estimation module in order to estimate the speed on the specified road on which the traffic condition is inferred. Since the sensors have varied positions on the map, some of them may have $pos_l^s$ located on the specified road, while others

**Algorithm 1** - Matching

**Input:** All cars $s_i$ participating in sensing services
**Output:** Set of targeted cars $S_{final}$ located in the specified destination
Construct a list $S_{init}$: Ø for the estimated targeted cars
**for** each sensor $s_i$ **do**
  get $s_i$ last position $pos_l^s$ from the server's database
  **if** $pos_l^s$ == onRoad || $pos_l^s$ == headingToRoad **then**
    **if** $s_i$ == available **then**
      **if** batteryLevel ≥ 20% **then**
        **if** reputation == high **then**
          **if** capability == true **then**
            **if** dataAcc == good **then**
              add $s_i$ to $S_{init}$
Construct a new list $S_{final}$ for the targeted cars
**for** each sensor $s_i$ in $S_{init}$ **do**
  send sensing request $r_i$ to $s_i$ and get its current position $pos_{cur}^s$
  **if** $pos_l^s$ == onRoad && $pos_{cur}^s$ == onRoad **then**
    add $s_i$ to $S_{final}$
  **if** $pos_l^s$ == onRoad && $pos_{cur}^s$ == outOfRoad **then**
    add $s_i$ to $S_{final}$
  **if** $pos_l^s$ == headingToRoad && $pos_{cur}^s$ == onRoad **then**
    add $s_i$ to $S_{final}$

located on its adjacent roads, as the set of collectors encompasses the cars heading toward the desired destination. Similarly, at time $t_2$, when the server sends the sensing request to the set of selected data collectors, the sensors' $pos_{cur}^s$ could either be on the specified road or on its adjacent one in case it left it. The distance of road traveled by sensor $s$ is denoted as $r_i(pos_l^s, pos_{cur}^s)$, which takes only the distance traveled within the two intersections of the road without the adjacent links as the data consumers ask for the condition of a specific road. Knowing $r_i(pos_l^s, pos_{cur}^s)$ of each sensor $s_i$ during the interval $(t_1, t_2)$, the server can compute their average speed vi defined by

$$v_i = \frac{r_i(pos_l^s, pos_{cur}^s)}{(t_1, t_2)} \tag{3}$$

The road condition represented by the mean speed is calculated according to equation (4) [36]

$$v_{mean}(t_2) = \frac{\sum_{s \in S_{final(t_2)}} [v_i \times r_i(pos_l^s, pos_{cur}^s)]}{\sum_{s \in S_{final(t_2)}} r_i(pos_l^s, pos_{cur}^s)} \tag{4}$$

where the mean speed $v_{mean}$ of a particular road at the request time $t_2$ is a function of the length of the road traveled and covered by each sensor s in the final set $S_{final}$, along with their average mobile speed. Note that this approach is widely used for traffic speed estimation, and works under the assumption that vehicles' speeds are constant. Typically, the ground truth ($v_{GT}$) is calculated using video surveillance of real traffic, which is a statistical measure that describes the entire traffic flow as follows

$$v_{GT}(t_k) = \frac{l}{\frac{1}{|C_i(t_k)|} \times \sum_{c \in C_i(t_k)} \triangle t_c} \tag{5}$$

We use visual observation of the traffic simulation to determine a set of cars $C_i(t_k)$ that enter the road segment within a certain time window $(t_1, t_2) \subseteq (t_k - \tau, t_k + \tau)$, where $t_k$ is the chosen moment in time to calculate the ground truth and $\tau$ is a predefined constant. For those set of cars, we calculated the time taken by each one of them ($\triangle t_c$) to traverse the road segment of length $l$. To determine the accuracy of the obtained results, we calculate the estimation error using equation (6) that represents the absolute value of the calculated mean speed minus the ground truth

$$\bar{E} = | v_{mean} - v_{GT} | \tag{6}$$

## 5. Prototype and Implementation

In order to validate our proposed solution, we combined prototyping with simulation traces generated using VanetMobiSim, which is a widely used traffic simulator that generates realistic vehicular movement traces, based on macroscopic and microscopic mobility models [37]. Instead of using real sensory data collected using phones, we opted for simulation traces as it allows the generation of a large set of data for our experiments and enables the control of different parameters (e.g. roads' topology, number of cars used, mobility model, and speed limits on the roads). In our experiments, we used a macroscopic mobility model that deals with properties such as traffic density, speed and flow.

### 5.1. Prototype software architecture

Figure 2 illustrates our prototype software architecture. The prototype, which was implemented in JAVA, consists of three main components: a data consumer node generating sensing trigger requests; a vehicular sensing platform node matching requests with collectors, managing the sensed data and estimating traffic status; and data collector nodes responding to the sensing requests and publishing their sensed data. Communication between the different components is achieved using REST APIs. To simplify the development of RESTful Web Services, we have selected the open source Jersey framework [27] that functions as a JAX-RS Reference Implementation [38], and Grizzly Application server that deploys the web services. Each component encompasses a PostgreSQL repository [39] to store the relevant sensed data.

As shown in Figure 2, the data consumer is a node consisting of a request/response handling module responsible of the generation of sensing requests and the handling of responses; a sensing session manager responsible of the tracking of the sensing sessions and their status; and a local sensing data repository (SDR) storing the collected data and the sensing sessions' statuses.

The vehicular sensing platform is the main node in our prototype. It consists of the following modules: a request/response handler responsible of the processing of received requests and responses; a validation and matching module implementing the matching algorithm and validating the data received; a request dispatcher and request queue responsible of queuing and dispatching requests to selected data collectors; a resource naming module responsible of assigning IDs to sensed entities, sensing services, and users; a publication engine handling voluntary data publications from data collectors; a traffic estimation module implementing the proposed traffic estimation algorithm; an analysis and reporting module responsible of the generation of advanced traffic reports from the collected data; and a sensing data repository (SDR) storing the sensed data, the generated

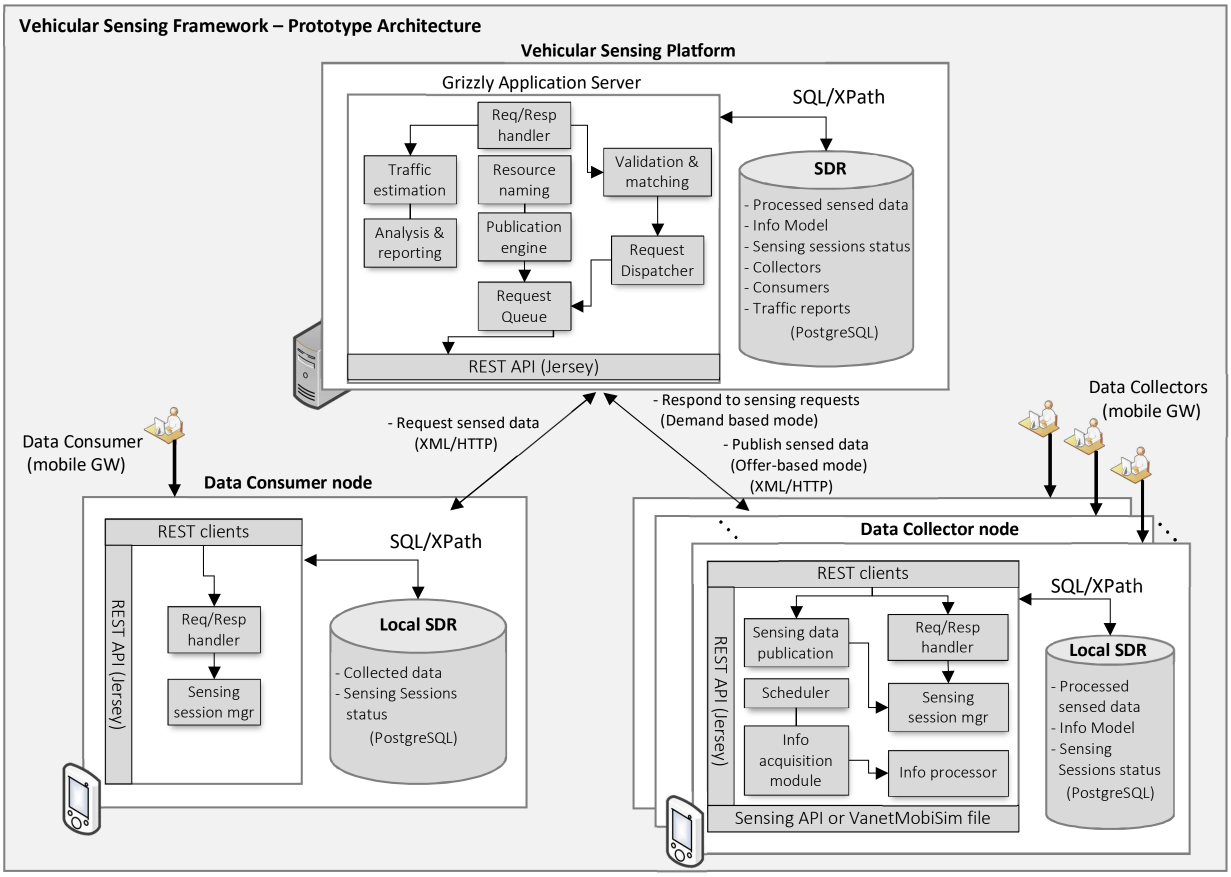



Figure 2: Prototype Components

traffic reports, the sensing sessions' status as well as information about data collectors and consumers.

Instead of hosting the data collector node logic on real mobile devices, we used instances of the data collector nodes running on one machine to simulate a large number of data collectors. Furthermore, we used VanetMobiSim to simulate different traffic conditions (i.e. the positions and speeds of the cars moving on the simulated roads), and stored this information in a file that was made accessible to the data collectors' instances. This file, which contains information related to all simulated car nodes, is initially processed by each data collector node to retrieve its specific information throughout the simulation lifetime, and stored on a local DB on the node in question. This combination of prototyping and simulated traffic data allows testing at different scales, not to mention the control of the traffic parameters that would not be possible with a real life prototype deployed on smart phones hosted in moving vehicles.
To achieve the functionality of a data collector, each data collector node consists of a request/response handling module responsible of receiving the sensing requests and publishing their sensed data; a sensing data publication module responsible of publishing the sensed data to the platform (either following a trigger or voluntarily); a sensing session manager module responsible of the tracking of the sensing sessions and their status; a scheduler module responsible of scheduling the processing of multiple requests received from the platform; an info acquisition module responsible of the retrieval of the car position and velocity sensed data (at that specific time instance) from the PostgreSQL local database (the SDR) hosted by the data collector node; an information processor module responsible of processing and formatting the messages exchanged via REST API between the vehicular sensing platform and the data collector node; and a SDR that stores all the data collector position/velocity information throughout the lifetime of the simulation, to be used whenever the platform asks for data collection. It should be noted that the information stored in the SDR is used either to publish data voluntarily to the sensing platform without any solicitation and trigger, or used to respond to sensing requests by sending the vehicle's velocity and position at certain time instance to the platform, thus covering two modes of information publication (trigger based publication and voluntary publication).

### 5.2. *Testbed Setup, Datasets, and Test Scenarios*

As shown in Figure 3, the experimental setup consists of three main components: One data consumer node triggering the sensing requests, one vehicular sensing platform node responsible of data and sensing requests/responses management and implementing the matching and traffic estimation algorithms, and one data collector management node that instantiates the needed data collector instances and dispatches sensing requests to the relevant ones. The used machines are equipped with Intel

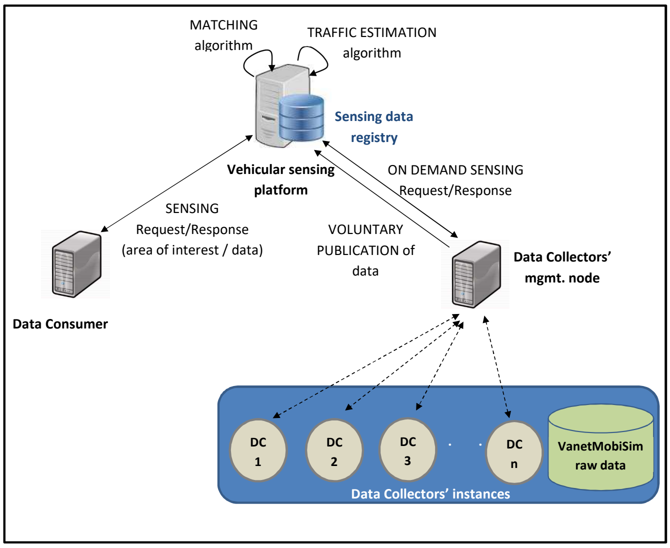



Figure 3: Testbed Setup

Core™2 Duo E6550, 2.33GHz processor and 4GB of RAM, 10000 RPM HDD, 100MBPS link and running Ubuntu 12.04 LTS.

To populate the raw data repository accessible to the data collector instances, four data sets were generated using VanetMobiSim simulations. The simulation runs were configured to simulated four traffic conditions, namely: free flowing, moderately congested, congested, and highly congested. Furthermore, in order to compare our proposed on-demand sensing approach to the traditional continuous sensing approach, two test scenarios were used in our experiments as illustrated in Figures 4 and 5.

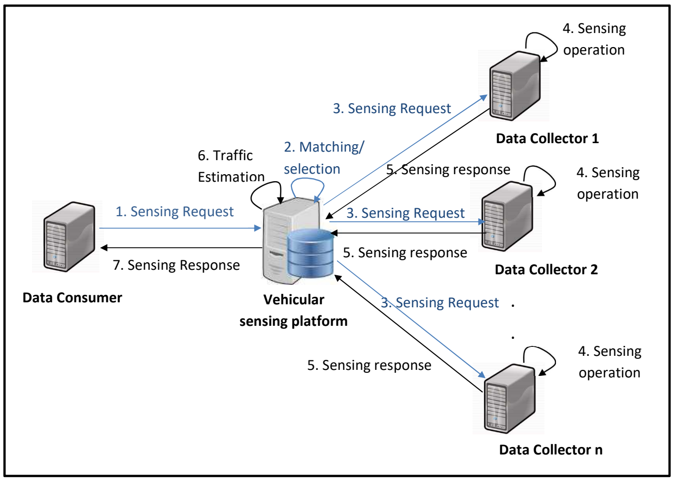



Figure 4: On Demand Sensing Scenario

In the on-demand sensing scenario depicted in Figure 4, the first interaction is triggered by the data consumer, which sends a sensing request to the vehicular sensing platform asking for the traffic condition in an area of interest (i.e. specific position or street). The sensing platform will then run the matching algorithm to get the list of suitable data collectors satisfying the matching criteria and forward to them the sensing request. Each data collector will perform the sensing operation (i.e. acquiring its position and speed in that case) and sends the sensed information as a response to the sensing platform. After receiving the responses from all targeted data collectors, the sensing platform will run the traffic estimation algorithm to estimate the traffic speed/condition. This information is then used to build a traffic report, which is sent by the sensing platform as final response to the data consumer.

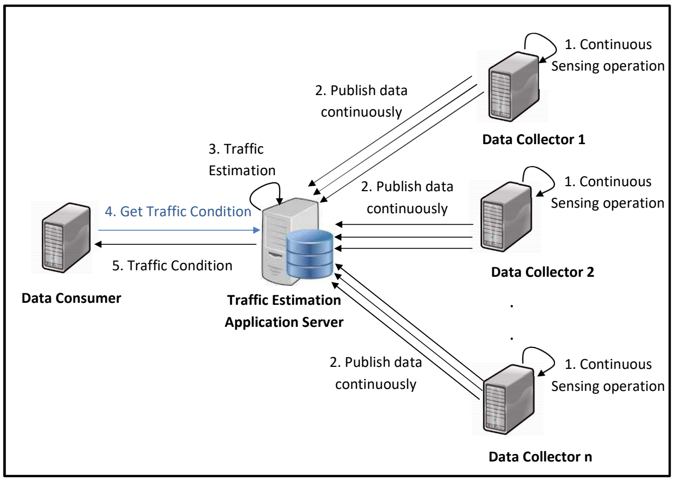



Figure 5: Continuous Sensing Scenario

In the continuous sensing approach depicted in Figure 5, the sensing operation is performed in a continuous fashion by all the data collectors, which publish their sensed information on a regular basis to the sensing platform. When a data consumer sends a sensing request to the platform, the latter uses the sensed information previously published to estimate the traffic condition using the traffic estimation algorithm, and then sends the final response (i.e. the traffic report) to the data consumer.

## 6. Experimental Results and Discussion

### *6.1. Performance Evaluation Strategy and Metrics*

The objectives of the experiments we conducted are to: (1) assess the performance of the two main algorithms implemented by the on-demand sensing platform (i.e. the matching and the traffic estimation algorithms); (2) evaluate the overall system performance including all the communications and processing overhead; and (3) compare the performance of the on-demand and continuous sensing approaches, using the two test scenarios presented in figures 4 and 5.

To achieve those goals, a number of testing approaches and performance metrics were used, as summarized in table 3. The detailed analysis of the conducted tests will be presented in the coming sections.

### *6.2. Algorithms' Performance Evaluation*

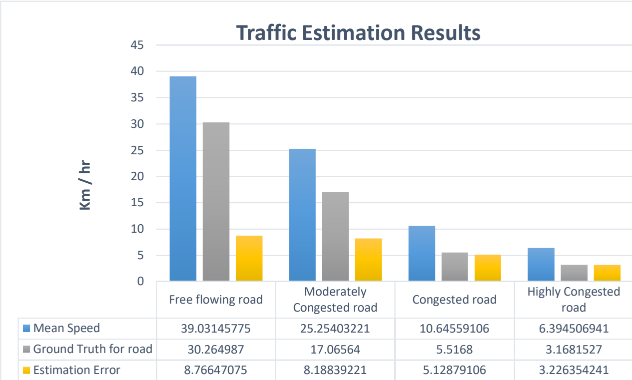
#### *6.2.1. Traffic estimation algorithm*

Figure 6 depicts the performance of our traffic estimation algorithm, when applied to four types of roads: a free flowing

| Test Category | Performance Metric | Description of how metric was measured/calculated | Test scenarios used |
|---|---|---|---|
| Traffic Estimation Algorithm | - Mean speed<br>- Ground truth<br>- Traffic estimation Error | Mean speed: calculated using equation 4<br>Ground truth: calculated using equation 5<br>Estimation error: calculated using equation 6 | Four scenarios were used:<br>1. Free flowing road 2. Moderately Congested road<br>3. Congested road 4. Highly Congested road. |
| Matching Algorithm | Response time | Time needed for the matching algorithm to return the set of selected cars located in the area of interest and matching the specified matching criteria. | Four variants of the matching algorithm were tested by varying the dataset (i.e. the # of cars processed during the selection) and the matching criteria used. The four variants are: 1. Six matching criteria (Proximity, availability, Data collection capability, accuracy, battery level, reputation) 2. Four matching Criteria (Proximity, availability, Data collection capability, accuracy) 3. Three matching Criteria (Proximity, availability, Data collection capability) 4. Two matching Criteria (Proximity & availability)<br>The dataset for each experiment was crafted in a way to show the difference between the different matching criteria. For instance, to highlight the effect of reputation as matching criteria, we introduced malicious nodes that injected wrong data in the data set nodes, which would be filtered out only if reputation is used as matching criteria. For accuracy, we introduced data that is rounded and not very accurate. This approach allows the differentiation between the different versions of the multi-criteria matching algorithm, and to show the trade-off between performance and accuracy. |
| | Matching error % | Matching error %: calculated as # of cars selected by algorithm / # of cars satisfying the matching criteria (calculated manually) * 100 | |
| System Load Testing | Response time | Time from when sensing request (msg. 1 in fig. 4) is sent until sensing response (msg. 7 in fig. 4) is received. | Using the on-demand sensing scenario presented in Fig. 4, we varied the number of requests sent by the data consumer to the platform from 1 to 2000 requests, and measured the systems response time and network load. |
| | Network Load | Size of packets exchanged for the end-to-end interaction (between sensing request and sensing response) | |
| System Data Frequency Based Testing - Continuous Sensing Approach | Response time | Time from when traffic condition request (msg. 4 in fig. 5) is sent until traffic condition response (msg. 5 in fig. 5) is received. | Using the continuous sensing scenario presented in Fig. 5, we varied the frequency of the voluntary data publications made by data collectors to the platform, as follows: each 30 secs, each 1 minute, each 5 minutes, and each 10 minutes. The following metrics were measured: response time, network load, and traffic estimation error. |
| | Network load | Size of packets exchanged for the end-to-end interaction (between voluntary publication of data and traffic condition response) | |
| | Traffic estimation error % | Calculated using equation 6 | |
| System Data Frequency Based Testing - On Demand Sensing | Response time | Time from when sensing request (msg. 1 in fig. 4) is sent until sensing response (msg. 7 in fig. 4) is received. | Using the on-demand sensing scenario presented in Fig. 4, we varied the number of received sensing requests/hour by the platform, as follows: 10 requests/h, 100 requests/h, 500 requests/h, 1000 requests/h, and 10,000 requests/h The following metrics were measured: response time, network load, and traffic estimation error. |
| | Network load | Size of packets exchanged for the end-to-end interaction (between sensing request and sensing response) | |
| | Traffic estimation error % | Calculated using equation 6 | |
| Participation % Based Testing | Response time | Time from when sensing request (msg. 1 in fig. 4) is sent until sensing response (msg. 7 in fig. 4) is received. | Using the on-demand sensing scenario presented in Fig. 4, we varied the % of cars participating in the sensing activity and to see the impact on the accuracy of the results. The % of targeted cars was varied from 100% of cars (continuous sensing case), to 70%, to 50%, to 30%, to 10%. The following metrics were measured: response time, network load, and traffic estimation error. |
| | Network load | Size of packets exchanged for the end-to-end interaction (between sensing request and sensing response) | |
| | Traffic estimation error % | Calculated using equation 6 | |
| Quality of Sensed Information Testing | Traffic estimation error | Calculated using equation 6 | Using the on-demand sensing scenario presented in Fig. 4, we varied the % of cars participating in the sensing activity (from 100% to 10%) as well as the matching criteria used (using 6, 4, 3, 2, and 1 matching criterion) in order to study the impact of the selection criteria on the Quality of sensed information & traffic estimation accuracy. The traffic estimation error was measured in that case. |

Table 3: Testing Strategies and Metrics

road, a moderately congested road, a congested road, and a highly congested road. For each scenario, we calculated the estimated mean speed, the ground truth for the road, and the traffic estimation error, as shown in the figure. By analyzing the obtained results, we notice that the mean speed estimation method yields more accurate results in the free flowing roads than in the more congested road, with an estimated mean speed of 39 Km/h on a road with a ground truth of 30.26 Km/h (for the free flowing case), vs. an estimated mean speed of 6.39 Km/h on a highly congested road with a ground truth of 3.16 Km/h. Another observation is that the mean speed method resulted in speed over-estimation in both congested and uncongested conditions. In absolute vehicular speed terms, the obtained traffic estimation results are very good since we are more interested in the traffic status (i.e. free flowing, moderately congested, congested, and highly congested) rather than the actual speed on the road. Thus, since the estimated traffic mean speed values were close to the ground truth values on the tested roads, the correct traffic condition was inferred in the four tested scenarios.




| | Free flowing road | Moderately Congested road | Congested road | Highly Congested road |
|---|---|---|---|---|
| Mean Speed | 39.03145775 | 25.25403221 | 10.64559106 | 6.394506941 |
| Ground Truth for road | 30.264987 | 17.06564 | 5.5168 | 3.1681527 |
| Estimation Error | 8.76647075 | 8.18839221 | 5.12879106 | 3.226354241 |

Figure 6: Traffic estimation results

### 6.2.2. *Matching algorithm*

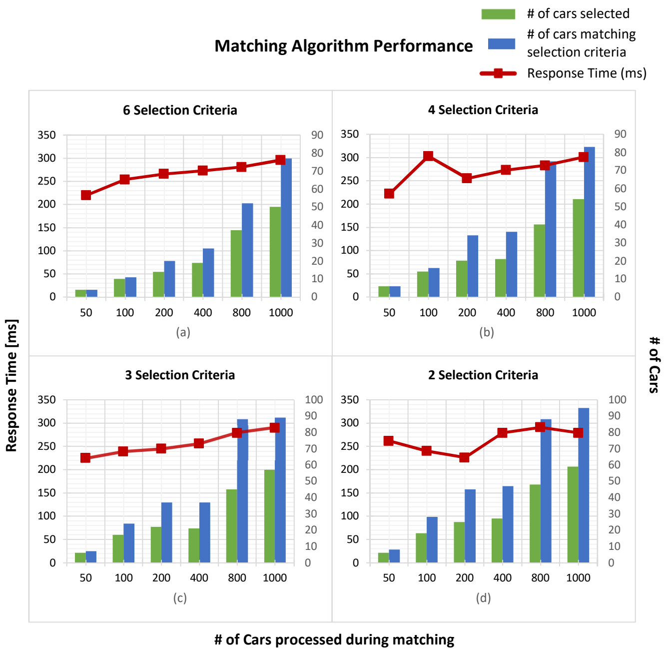



Figure 7: Performance for six/four/three/two criteria of matching algorithm

Figure 7(a) shows the performance of the multi-criteria matching algorithm, when all six selection criteria (i.e. availability, proximity, data collection capability, accuracy, battery level, and reputation) are used for participants' selection. In this experiment, the number of cars on the road, which were processed during the matching varied from 50 cars to 1000 cars. As expected, when the number of cars available on the road increased (i.e. the size of the dataset increased), the number of targeted collectors matching the selection criteria increased. For instance, when the number of cars on the road is 50, the number of selected data collectors is 4, while when there were 1000 cars on the road, the number of targeted collectors rose to 80. Examining the performance of the 6-criteria matching algorithm, we observe that the algorithm yielded good results, by selecting 4 out of 4 eligible cars for a dataset of 50 cars (i.e. matching error percentage of 0%), and 10 out of 11 eligible

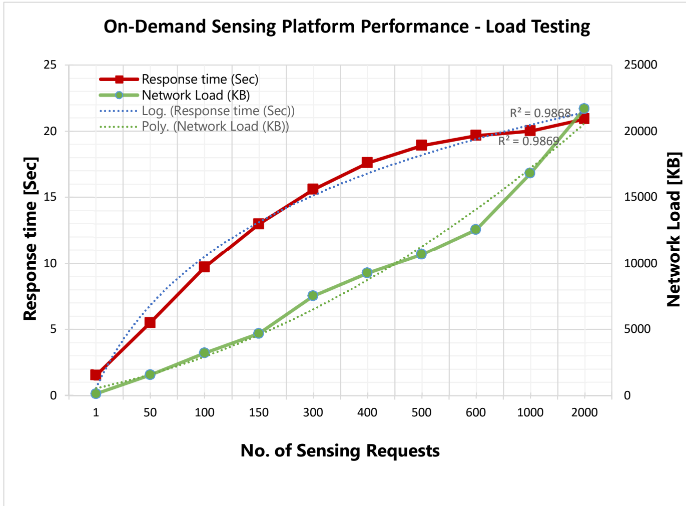



Figure 8: Load testing results for on-demand sensing platform

cars for a dataset of 100 cars (i.e. matching error percentage of 10%). We notice that the matching error % increases with an increase of the size of the cars datasets processed. For instance, when the dataset consisted of 800 cars, the number of selected cars was 37 out of 52 eligible cars (i.e. a matching error percentage of 29%). As for the matching algorithm's response time, it varied between 220 ms in the case of 50 processed cars to 296 ms when 1000 cars were processed, which is an acceptable performance that would bear minimum impact on the end-to-end system response time. The other variants of the multi-criteria matching algorithm exhibit a similar performance from response time and matching error %, as shown in figure 7(b), (c) and (d). **It should be noted that in some scenarios, small delay from the data collectors' side may occur when specifying their locations and sending their responses back to the server. However, not to affect the waiting time of the requester, we set a threshold for the collectors' responses and we neglect any reply that comes after. Such delay in figure 7(b) is around 65 milliseconds, which is considered relatively small.**

### 6.2.3. *System's Performance Evaluation*

A. Load Testing:

In order to evaluate the behavior of the on-demand sensing system under variable loading conditions, we conducted some load tests using the test setup shown in figure 4. Figure 8 shows the obtained load testing results.

As shown in figure 8, the on-demand sensing system shows a logarithmic growth pattern in terms of response time, which ranged from 1.52 s for 1 sensing request to 20.9 s for a 2000 sensing requests. The response time is affected by four main operations related to on-demand sensing, namely: the multi-criteria participants' selection process; the waiting time required to receive sensed data from the targeted participants; the concurrent access to platform's DB for storage of different pieces of sensed data; the traffic estimation process and generation of traffic reports. The logarithmic growth pattern can be explained by the fact that at the beginning, data collectors need to be contacted to satisfy the sensing requests. However, as the number of sensing requests related to a certain area increases, the need for contacting data collectors diminishes, since fresh data is already available in the system and can be used directly for traffic estimation.

As for the generated network load, it showed a polynomial (quadratic) growth pattern with values ranging from 120 KB for 1 sensing request to 21689 KB for 2000 sensing requests. The network load's growth pattern can be explained by the fact that the more sensing requests are received, the more data collectors are targeted which multiplies the number of messages exchanged through the system.

B. Data-Frequency Based Testing in On-Demand vs Continuous Sensing:

In order to compare the on-demand and continuous sensing approaches, we conducted data frequency based testing in which we varied the sensing frequency (i.e. the number of sensing requests received per hour by the on-demand sensing platform and the number of voluntary publications made in continuous sensing mode) and measured the response time and network load generated in both cases. The presented results in Figures 9 and 10 illustrate clearly the benefits of the proposed on-demand approach in terms of traffic overhead and network load compared to the continuous, while maintaining very close traffic estimation accuracy in both of them.

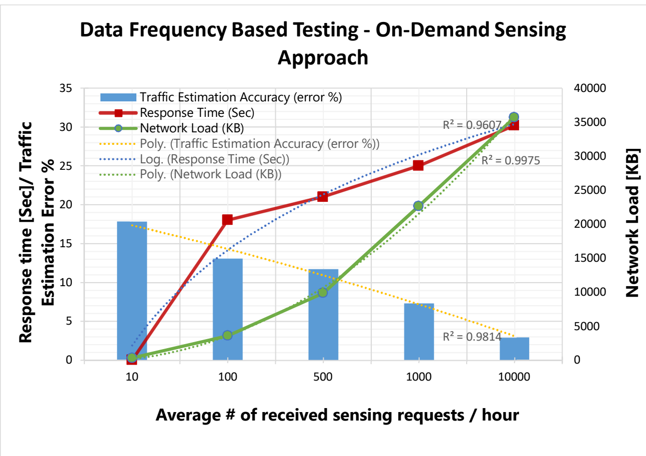



Figure 9: Data frequency based testing results for on-demand sensing approach

Figure 9 shows the data frequency based testing results for the on-demand sensing approach in terms of network load, response time, and traffic estimation accuracy (error %) as a function of the average number of received sensing requests/hour. As shown in the figure, the on-demand sensing system shows a logarithmic growth pattern in terms of response time, which increases from 15.264 s to 30.165 s as the average number of received sensing requests per hour increases from 10 to 10000. This growth pattern can be explained by the fact that the more requests are received per hour, the more fresh data is available in the platform, which can be reused to answer subsequent requests without the need to resort to data collectors. Furthermore, in some cases, the current traffic status reports may be already available in the system due to many requests in the same area, which will decrease the response time to the new data consumers requesting traffic conditions in the same area. On the other hand, the system's network load exhibits a polynomial (quadratic) trend line, ranging from 340 KB for an average of 10 sensing requests received/hour up to 3568 KB for an average of 10000 sensing request received/hour. This polynomial increase is attributed to the additional number of data collectors required for new requests, thus generating additional traffic load. As for the traffic estimation accuracy, we notice that as the average number of sensing received by hour increases, the traffic estimation error % decreases, dropping from 17.82% estimation error for 10 sensing requests received/hour to 2.9% estimation error for 10000 sensing requests received/hour. This can be explained by the fact that the more requests are received, the more data points are collected about a certain area, and the more accurate the traffic estimation results will be.

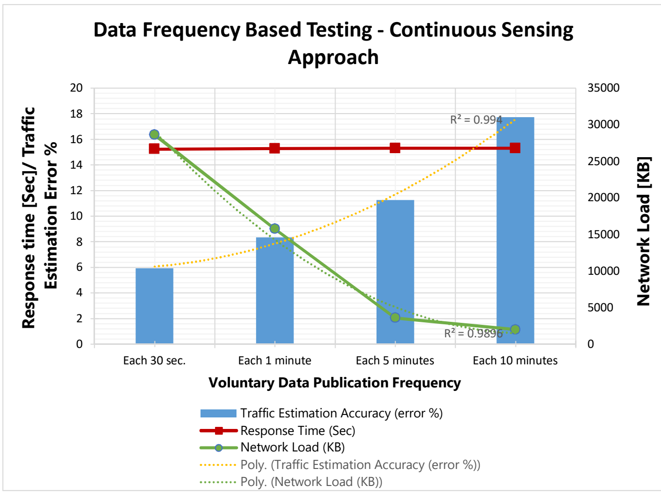



Figure 10: Data frequency based testing results for continuous sensing approach

On the other hand, Figure 10 shows the data frequency based testing results for the continuous sensing approach in terms of network load, response time, and traffic estimation accuracy (error %) as a function of the voluntary data publication frequency. Similar to the on-demand sensing case, in the continuous sensing case, we notice that the traffic estimation error decreases with the increase of the voluntary data publication frequency, dropping from an estimation error of 17.7% for data voluntarily published each 10 minutes to an estimation error of 5.9% for data published each 30 seconds. On the other hand, the network load follows a polynomial (quadratic) growth curve, which is expected with the increase in the number of data publication messages associated with an increased publication frequency (i.e. from each 10 minutes to each 30 seconds) . Finally, we notice that the response time remains constant with respect to the voluntary data publication frequency. This is due to the fact that when the platform receives a traffic condition request, it uses the data previously published in the system to estimate the traffic and send the final response. Therefore, the data publication frequency bears no effect on the response time in the continuous sensing case.

### C. Participation Based Testing in On-Demand vs Continuous sensing:

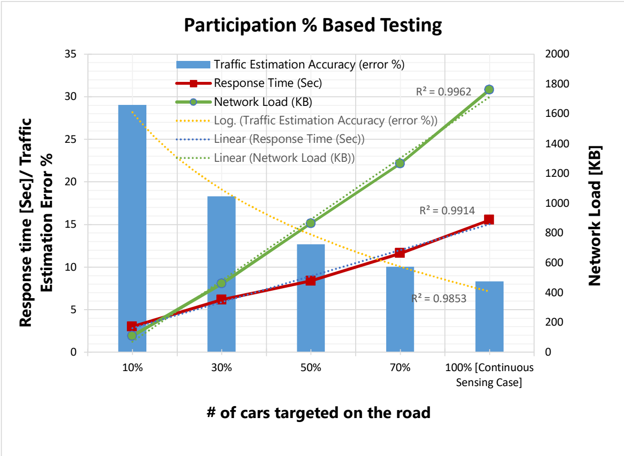



Figure 11: **Participation based testing results for On-Demand vs Continuous approaches**

The continuous sensing approach can be considered as a special case of on-demand sensing approach, in which data is acquired on a regular basis from a 100% of the cars, instead of occasionally from some of the cars. In order to test the impact of the percentage of cars participating in the sensing activity on the accuracy of the traffic estimation results, we carried a test in which the % of participating cars is varied from 10% to 100% and measured the response time, network load, and traffic estimation error. Figure 11 depicts the obtained results.

As observed, both the network load and response time increase in a linear fashion with the increase in the % of cars participating in the sensing activity. For instance, the network load and response time respectively achieved for 10% of cars targeted are 110 KB and 3 s. For a 100% of cars targeted (i.e. the continuous sensing case), the network load and response time increased to 1760 KB and 15.5 s. This is an expected result as more participation results in more messages exchange (i.e. higher network load) and more time to process those messages (i.e. higher response time). On the other hand, we notice that the traffic estimation error decreases in a logarithmic fashion, with the increase in the % of participating cars. This is due to the fact that the more cars are targeted, the more data points are collected about a certain area, and the more accurate is the traffic estimation result. It should however be mentioned that there is a compromise between the accuracy of the traffic estimation result needed and the system's performance in terms of network load and response time. In fact, a higher traffic estimation accuracy will be associated with a poorer system performance in terms of network load and response time. For instance, with a 100 % of cars targeted, we obtain the lowest traffic estimation error (i.e. 8.3%) along with the highest network load (i.e. 1760 KB) and the highest response time (i.e. 15.5 s). Decreasing the percentage of car participation to 50% results in penalty of 4% of additional traffic estimation error, but an improvement of 50.9% in terms of network load and an improvement of 46.5% in terms of response time. In the case of 30% participation rate, the additional traffic estimation error accrued is 10%, while the improvement in terms of network load is 73.8% and the improvement in terms of response time is 60.3%. Moreover, it is worth to mention about the high accuracy achieved by the proposed on-demand approach, where in all the cases the traffic estimation error is acceptable in order to determine the traffic status of the road, especially 30% and above, where the error rate starts to be similar to the continuous approach.

### D. Impact of Selection Criteria on Quality of Sensed Information and Traffic Estimation Accuracy:

In order to evaluate the impact of the participants' selection criteria on the traffic estimation accuracy, we varied both the % of cars targeted for a sensing activity as well as the # of criteria used for participants' selection from the ones targeted. Figure 12 depicts the obtained results.

As expected, for the 5 sets of matching/selection criteria used (i.e. 6 matching criteria, 4 matching criteria, 3 matching criteria, 2 matching criteria, and 1 matching criterion), increasing the % of cars targeted for sensing has a positive impact on the traffic estimation accuracy. Furthermore, when the same % of cars are targeted and the different variants of the matching approach are compared, the more selection criteria we use, the more accurate is the traffic estimation result. As shown in the figure, the 6-criteria matching approach (the blue curve) outperforms all other approaches (i.e. 4 criteria, 3 criteria, 2 criteria, 1 criteria) for all % of participating cars used. This is due to the fact that for the same % of cars targeted, the 6-selection criteria approach selects the best candidates yielding the highest quality records satisfying multiple quality of information criteria. Although the other approaches select the same number of candidates in each test scenario, the selected candidates provide lower quality information since some of the quality criteria are not considered in the selection process, thus yielding less accurate traffic estimation results.

It is very important to mention that even with less % of targeted cars, the selection approaches with more criteria outperform those with less selection criteria targeting a higher % of cars in some cases. As an example illustrated in the blue dotted area in Figure 12, the 6-selection criteria approach achieves a traffic estimation error percentage of 29.015% with only 10% of targeted cars, which is a lower traffic estimation error than the ones achieved by the 2-criteria approach and 1-criteria approach targeting 30% and 50% of the cars (yielding traffic estimation errors ranging between 30.29% and 38.84%). The same applies when comparing the 6-criteria approach targeting 30% of cars, to all other variants targeting 50%, 70%, and even 100% of cars (see green dotted area in the figure). This implies that using the 6 criteria approach and targeting 30% cars as candidates' yields more accurate results than targeting 100% of cars with only 4 selection criteria. We can therefore conclude that there exists

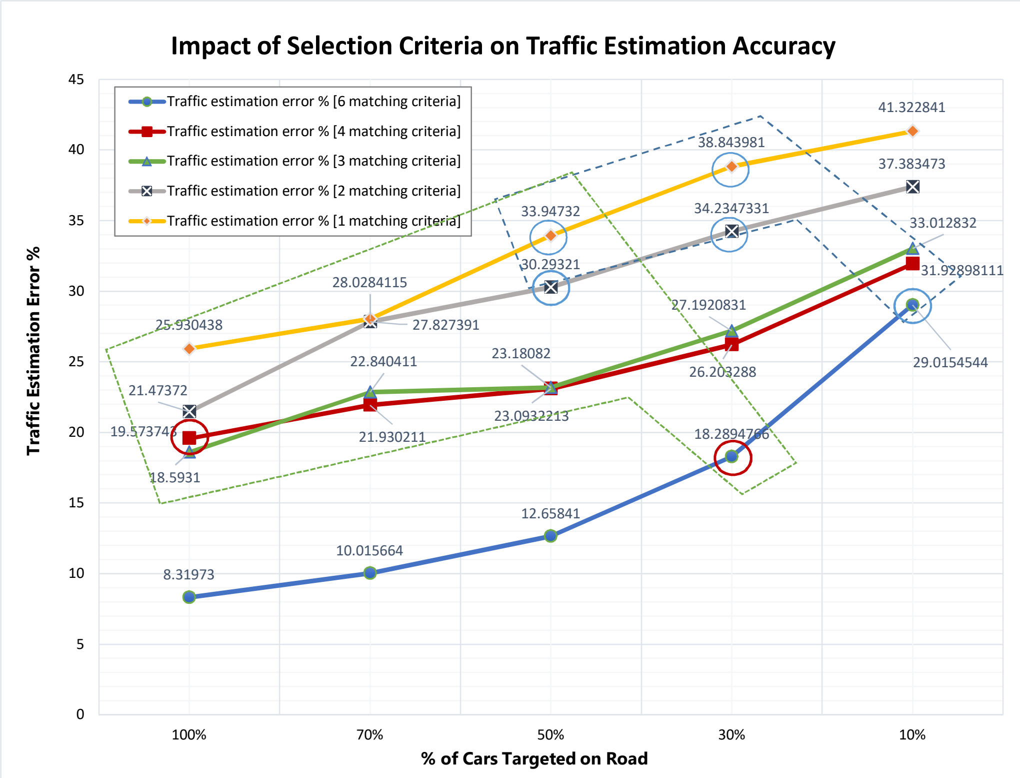



Figure 12: Impact of Selection Criteria on Traffic Estimation Accuracy

a trade-off between the number of selection criteria (i.e. the complexity of the selection approach) and the accuracy of the traffic estimation results obtained. Therefore, when more contextual information is available and multiple selection criteria can be used for participants' selection, a small % of participating cars can be targeted while achieving high traffic estimation accuracy. On the other hand, in the case of lack of availability of contextual information and the inability to use multiple selection criteria, a larger % of participating cars need to be targeted to compensate for the lower quality data records and maintain good traffic estimation accuracy. Based on the tests we conducted and the results obtained, we can conclude that using 6 matching criteria and 30% of targeted cars achieves the best trade-off in the test scenario and environment we used.

## 7. Conclusion

In this paper, we have proposed a novel infrastructure-less vehicular sensing framework enabling the on-demand sensing of traffic conditions, about any area of interest, by relying on a select set of mobile phone owners acting as data collectors. RESTful web service communication interfaces were defined to enable the communication between the sensing platform and the users. Furthermore, a multi-criteria participants' selection model and a mean-speed based traffic estimation model were proposed to support the operation of the vehicular sensing platform. The framework architecture was fully implemented using a combination of prototyping and traffic simulation traces generated using VanetMobiSim.

The obtained experimental results explore the benefits that can be offered by an on-demand participatory sensing approach in terms of achieving high traffic estimation accuracy and resource efficiency, when compared to the traditional opportunistic continuous sensing approach. Among the lessons learned from this work, we note the following: The on demand sensing is able to successfully infer the traffic status category regardless of the minor variation in the mean speed compared to the ground truth. Furthermore, we observed that more complex participants' selection approaches that rely on contextual information to select the best participants (offering the highest quality sensing data records) can yield a high traffic estimation accuracy, even with a low percentage of vehicles participating in the sensing activity. On the other hand, if this contextual information is not available and simpler participants' selection approaches must be used, then a higher percentage of participants' vehicles must be employed to compensate for the lower quality in the sensed data records and maintain a high traffic estimation accuracy. Those results clearly demonstrate that the proposed on-demand participatory sensing approach can achieve high traffic estimation accuracy, while maintaining a good system's performance in terms of reduced response time and network load.

## Acknowledgments

This work has been supported by the Associated Research Unit of the National Council for Scientific Research, CNRS-

Lebanon, Zayed University, and the Lebanese American University (LAU).

## References


[1] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications magazine*, vol. 48, no. 9, 2010.

[2] G. Dimitrakopoulos and P. Demestichas, "Intelligent transportation systems," *IEEE Vehicular Technology Magazine*, vol. 5, no. 1, pp. 77–84, 2010.

[3] B. Tian, Q. Yao, Y. Gu, K. Wang, and Y. Li, "Video processing techniques for traffic flow monitoring: A survey," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, 2011, pp. 1103–1108.

[4] D. Cerotti, D. Salvatore, G. Merlino, and A. Puliafito, "A crowd-cooperative approach for intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, pp. 1529–1539, 2017.

[5] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, 2011.

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

[7] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: rich monitoring of road and traffic conditions using mobile smartphones," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*. ACM, 2008, pp. 323–336.

[8] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, "Connected vehicles: Solutions and challenges," *IEEE internet of things journal*, vol. 1, no. 4, pp. 289–299, 2014.

[9] C.-M. Huang, M.-S. Chiang, D.-T. Dao, P. H.-M. , S. Xu, and H. Zhou, "Vehicle-to-infrastructure (v2i) offloading from cellular network to 802.11p wi-fi network based on the software-defined network (sdn) architecture," *Vehicular Communications*, pp. 288–300, 2017.

[10] M. Fountoulakis, N. Bekiaris-Liberis, C. Roncoli, I. Papamichail, and M. Papageorgiou, "Highway traffic state estimation with mixed connected and conventional vehicles: Microscopic simulation-based testing," in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 2016.

[11] W. Fawaz, R. Atallah, C. Assi, and M. Khabbaz, "Unmanned aerial vehicles as store-carry-forward nodes for vehicular networks," *IEEE Access*, vol. 5, pp. 23 710–23 718, 2017.

[12] https://feeneywireless.com/152-million-connected-vehicles-by-2020/.

[13] "The statistics portal," https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/, 2016.

[14] O. Abdel Wahab, A. Mourad, H. Otrok, and J. Bentahar, "Ceap: Svm-based intelligent detection model for clustered vehicular ad hoc networks," *Expert Systems With Applications*, vol. 50, pp. 40–54, 2016.

[15] W. AlRahal AlOrabi, S. Abdul Rahman, M. El Barachi, and A. Mourad, "Towards on demand road condition monitoring using mobile phone sensing as a service," in *The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016)*, vol. 83. Procedia computer science, 2016, pp. 345–352.

[16] R. Mizouni and M. El Barachi, "Mobile phone sensing as a service: Business model and use cases," in *Next Generation Mobile Apps, Services and Technologies (NGMAST), 2013 Seventh International Conference on*. IEEE, 2013, pp. 116–121.

[17] D. G. Adam and I. Oikonomidis, "Tarifa: Traffic and abnormalities road instructor for anyone," in *2nd Student Workshop on Wireless Sensor Networks*, 2009.

[18] R. Herring, A. Hofleitner, D. Work, O.-P. Tossavainen, and A. M. Bayen, "Mobile millennium-participatory traffic estimation using mobile phones," in *CPS Forum, Cyber-Physical Systems Week 2009*, 2009.

[19] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, "Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2009, pp. 85–98.

[20] J. C. Herrera, D. B. Work, R. Herring, X. J. Ban, Q. Jacobson, and A. M. Bayen, "Evaluation of traffic data obtained via gps-enabled mobile phones: The mobile century field experiment," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 4, pp. 568–583, 2010.

[21] Z. He, B. Cao, and Y. Liu, "Accurate real-time traffic speed estimation using infrastructure-free vehicular networks," *International Journal of Distributed Sensor Networks*, 2015.

[22] X. J. Ban and M. Gruteser, "Towards fine-grained urban traffic knowledge extraction using mobile sensing," in *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*. ACM, 2012, pp. 111–117.

[23] Y. X. Wazir Zada Khan, M. Y. Aalsalem, and Q. Arshad, "Mobile phone sensing systems: A survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 402–427, 2013.

[24] T. Das, P. Mohan, V. N. Padmanabhan, R. Ramjee, and A. Sharma, "Prism: platform for remote sensing using smartphones," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM, 2010, pp. 63–76.

[25] B. Płaczek, "Selective data collection in vehicular networks for traffic control applications," *Transportation Research Part C: Emerging Technologies*, vol. 23, pp. 14–28, 2012.

[26] "Ericsson mobility report," https://www.ericsson.com/assets/local/mobility-report/documents/2017/ericsson-mobility-report-june-2017.pdf, 2017.

[27] "Jersey restful web service," http://jersey.java.net/.

[28] F. Hao, M. Jiao, G. Min, and L. T. Yang, "A trajectory-based recruitment strategy of social sensors for participatory sensing," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 41–47, 2014.

[29] H. Li, T. Li, and Y. Wang, "Dynamic participant recruitment of mobile crowd sensing for heterogeneous sensing tasks," in *Mobile Ad Hoc and Sensor Systems (MASS), 2015 IEEE 12th International Conference on*. IEEE, 2015, pp. 136–144.

[30] Z. Song, E. Ngai, J. Ma, X. Gong, Y. Liu, and W. Wang, "Incentive mechanism for participatory sensing under budget constraints," in *Wireless Communications and Networking Conference (WCNC), 2014 IEEE*. IEEE, 2014, pp. 3361–3366.

[31] Z. Song, C. H. Liu, J. Wu, J. Ma, and W. Wang, "Qoi-aware multitask-oriented dynamic participant selection with budget constraints," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 9, pp. 4618–4632, 2014.

[32] Z. Song, B. Zhang, C. H. Liu, A. V. Vasilakos, J. Ma, and W. Wang, "Qoi-aware energy-efficient participant selection," in *Sensing, Communication, and Networking (SECON), 2014 Eleventh Annual IEEE International Conference on*. IEEE, 2014, pp. 248–256.

[33] D. Peng, F. Wu, and C. Guihai, "Pay as how well you do: A quality based incentive mechanism for crowdsensing," in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, 2015, pp. 177–186.

[34] Y. Wen, J. Shi, Q. Zhang, X. Tian, Z. Huang, H. Yu, Y. Cheng, and X. Shen, "Quality-driven auction-based incentive mechanism for mobile crowd sensing," *IEEE Transactions on Vehicular Technology*, vol. 64, pp. 4203–4214, 2015.

[35] D. Yang, G. Xue, X. Fang, and J. Tang, "Incentive mechanisms for crowdsensing: Crowdsourcing with smartphones," *IEEE/ACM Transactions on Networking*, vol. 24, pp. 1732–1744, 2016.

[36] X. Li, W. Shu, M. Li, H.-Y. Huang, P.-E. Luo, and M.-Y. Wu, "Performance evaluation of vehicle-based mobile sensor networks for traffic monitoring," *IEEE transactions on vehicular technology*, vol. 58, no. 4, pp. 1647–1653, 2009.

[37] "Vanetmobisim traffic simulator," http://vanet.eurecom.fr/.

[38] "Java api for restful services," http://jax-rs-spec.java.net/.

[39] "Postgresql database management system," http://www.postgresql.org/.