2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing

Measuring Distances with RSSI from Vehicular Short-Range Communications

Roberto S. Yokoyama*, Bruno Y. L. Kimura†, Leandro A. Villas*, and Edson D. S. Moreira‡
*Instituto de Computação – Universidade Estadual de Campinas
†Instituto de Ciência e Tecnologia – Universidade Federal de São Paulo
‡Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo
sadao@lrc.ic.unicamp.br, bruno.kimura@unifesp.br, leandro@ic.unicamp.br and edson@icmc.usp.br

***Abstract*—In vehicular networks many applications and services require the use of vehicle position, which is typically obtained from off-the-shelf Global Positioning Systems (GPS). However, GPS' position accuracy is low and the GPS receiver relies on line-of-sight signalling from the satellites. For the sake of application consistency, when the GPS position is not received due to signalling failures, vehicular applications need to find a way to obtain their positions or, at least, their distances to other vehicles or road-side units (RSU). In this paper, we propose an architecture based on beaconing vehicular communication and its received signal strength indicator (RSSI) to estimate the distance from the vehicle to the RSU and vice-versa, as base for an ancillary positioning system. To this end, we implemented and deployed a vehicular network testbed in order to collect datasets of RSSI and distances. Thus, we characterized the RSSI behaviour and evaluated our architecture using real environment datasets. In the absence of GPS signalling, the results show that RSSI-based distance estimation provides absolute mean error of $\approxeq 16$ m for the vehicle and $\approxeq 14$ m for the RSU. Finally, we compared the solution with the traditional distance measure based on free-space model.**

I. Introduction

Several applications in vehicular networks rely on geographical location information to compute the distances between vehicles or infrastructure around [4]. Currently, the most comprehensive positioning tools for the vehicle's position is the Global Positioning System (GPS). However, GPS needs line-of-sight to receive signals from the satellites, otherwise, the position is inaccurate or absent in situations where the satellite signal is obfuscated as in tunnels, places with tall buildings, streets with high density of trees, region with excessive air pollution, or simply bad weather, e.g. rainy days or snowstorms. Moreover, there is security problems, since the GPS civil channel is not encrypted, so the signal can be manipulated to modify the vehicle position [2]. Thus, we can not assume that the GPS hardware/software is free from communication problems.

Vehicular applications, mostly safety applications, require another way to estimate the distances of nearby vehicles or infrastructure in order to preserve their consistency when the position is inaccurate or absent. Based on the distances, vehicle can infer its position and become aware of the environment [3]. In the literature, there are several proposals for measuring distance between vehicles using different technologies [1]. Ying *et al*. [8] has proposed the use of digital image processing to measure the distance from the car ahead up to 100 m, with accuracy of 7 m, which is calculated in 400 ms. However, image processing is limited to the camera view and the position of vehicles ahead. Today, modern cars are fitted with infrared sensors to measure distances to objects or obstacles, but also suffer from the line-of-sight problem and inaccuracy measurements due to the interferences, e.g. engine smoke [5]. Regarding vehicle networks, Widmann *et al*. [7] present a recent work with *zigbee* technology embedded in sensors to compare the RSSI and time of flight (ToF) techniques to measure distances between vehicles within tunnels. In real environment, the errors on average are 175 m for RSSI and 67.7 m for ToF in interval distances from 100 to 200 m.

We propose an architecture to estimate the distance between the vehicles to RSU and the RSU to vehicle based on RSSI. The idea is to take advantage of the vehicular dedicated short-range communication (DSRC) by beaconing, which is defined as standard of vehicular networks [9], in order to collect information of RSSI and estimate the distance in situations of GPS device failure. This approach has the advantages: *i)* estimated distances above 100 m, independent of the vehicle direction or RSU; *ii)* it is fully distributed; and *iii)* it does not depending on a minimum amount of compliant nodes that implement the solution. On the other hand, the estimated position accuracy is around dozen meters. Our solution is able to estimate distances by using only one reference point (vehicle or RSU) as a source of RSSI, whose vehicle wants to measure the distance. This solution is useful for researchers and engineers to design applications for vehicles using distances with precision of a few meters.

To analyse the RSSI behaviour through statistical techniques, we used real environment data collected from experiment executed in a vehicular networks testbed. This takes the advantage of the RSSI data were under ordinary interferences such as the metallic parts of the vehicle, engine noise, solar radiation, vehicle movement, etc. To estimate the distance, we implemented an architecture into two phases: first is the collection and storage of RSSI data, and second is the inference based on the processed and stored data represented by instances. The architecture was validated and evaluated through the distance estimation of the vehicle to the RSU and the RSU to the vehicle. As we expected, the results show that the position mean error is of at least 14 m, which is usually greater than the one obtained from simulations [10]. In comparison with the free-space model to obtain the distance based on the RSSI, the solution is better than free-space for distances higher than 120 m due the error of distance measure reminders close to the absolute average error, while the free-

978-1-5090-0154-5/15 $31.00 © 2015 IEEE
DOI 10.1109/CIT/IUCC/DASC/PICOM.2015.17

TABLE I. TABLE OF SYMBOLS

Symbol	Description
$\triangle$	Distance interval
v	Vehicle
$\bar{d}$	Estimated distance
f	Frequency of the vehicle sends beacons
r_j	*j*-th infrastructure or vehicle who sends beacons
n	Total of instance stored in the vehicle
$\hat{I}_n$	*n*-th instance
$\Bbbk$	Technique used to compare instances
$\bar{X}$	Average of RSSI
$\bar{s}$	Standard deviation of RSSI
R	Amplitude of RSSI
m	Sample size to make one instance
k	Number of selected neighbouring instances from k-NN
τ	Percentage of the matches with distance estimated and distance interval
ε	Absolute mean error
C	Attenuation signal constant in dBm

space error tends to increasing.

The reminder of this paper is organized as follows: Section II describes the application, hardware, testbed and the data collection; Section III describes the characterization of the RSSI; Section IV formalizes the problem concerned in this paper; Section V presents the architecture proposed; Section VI shows the validation and evaluation; finally, Section VII discusses the conclusions.

II. EXPERIMENTAL PLATFORM

The study conducted is entirely based on the analysis of datasets collected from experiments in real environment, i.e. in a vehicular network testbed. Subsection II-A introduces the beaconing application middleware. Subsection II-B presents the testbed hardware. Subsection II-C discusses the testbed deployment. Finally, Subsection II-D details the methodology used to collect the measurements.

A. Beaconing Application

We implemented a beaconing application as a middleware between the Application layer and MAC layer to provide facilities for manipulating the native 802.11 beacon frame, especially for inserting new data and controlling the beacon interval broadcasting. Such a middleware was implemented with C Language and its code was compiled for Linux Operating System OpenWRT blackfire 10.0.3. System Calls (syscall) from ordinary C APIs are utilized to set up the communication parameters, such as RF channel, transmission power and bit-rate. The transmission interval is controlled by the own application with a timer syscall.

The middleware has two main procedures: *1)* send beacon messages, and *2)* receive beacon messages. These procedures run in the network nodes, vehicle and RSU, to broadcast beacons simultaneously.

1) Send beacon: Within IEEE 802.11 beacon frame is available for manipulating up to 239 data bytes of IE (Information Element) Vendor field. Thus, application fills out specific data structure which is accommodated in the IE Vendor field. In general, this procedure identifies the Element ID (value 221) of the IE Vendor field and replaces the bytes in sequence with the ones of the application data structure. SSID (Service Set Identifier) field is also modified to broadcast a application beacon. This allows to identify desired beacons of the testbed network. To obtain the geographic position, the middleware connects to the daemon process of GPS (GPSD 2.4 Daemon) with specific socket descriptor. The GPSD returns the position from the GPS receiver though that socket. Then, the beacon is built and sent by using the $send$ function over another descriptor, which is a raw socket descriptor. Raw sockets allow packet transmission from the application directly to the link layer without passing through the TCP/IP protocol stack.

2) Receive beacon: Beacon reception is implemented with PCAP (Packet Capture) API. PCAP works with the wireless interface in promiscuous mode and captures all kinds of packets that are transmitted in the RF channel. To avoid application overhead of packet selection, we use PCAP filter to capture only the beacons packets. Once received, beacons are processed according to the SSID of the testbed network. Data carried in the beacon are processed similarly to the procedure of sending. The frame is parsed to locate the Element ID of IE Vendor and, then, the data struct is read on it. Thus, the timestamp, position and speed of the receiver are recorded in the log file.

B. Hardware

The main hardware used to deploy the testbed is described in Table II. The Alix motherboard was configured as a wireless router with Linux OpenWRT to work as both on-board unit (OBU) and road-side unit (RSU). Alix motherboard provides no support for IEEE 802.11p vehicular standard [6]. To allow nearest operation to the vehicular standard, such a router was equipped with IEEE 802.11a wireless card that works at 5GHz, the same as the IEEE 802.11p. Also, we use antenna with magnetic docking, GPS receiver and battery for independent power supply, all plugged into router.

1) OBU Configurations: To prepare the vehicle's OBU, we fasten the wi-fi antenna and GPS by using magnetic base on the vehicle's roof. The power source of the OBU's router is a battery of $12V$ connected to it via PoE. Router and battery are fasten inside the vehicle. Figure 1(a) is a picture of the vehicle with antenna and GPS receiver on its roof. As can be seen, cables pass through the window to reach the OBU's router inside the vehicle. Lastly, the OBU's router is accessed remotely via Secure Shell (SSH) from a notebook, so that it is possible to start, monitor and finish the beaconing application.

2) RSU Configurations: A steel rod of 1.5 m in length is used to stand the RSU, as shown in Figure 1(b). The antenna and the GPS receiver are positioned at the top of the rod by the magnetic base. Before initializing the application in the RSU's router via remote access from the notebook, the SSH session is supported with Screen program, which allows to disconnect the network cable from the notebook without shutdowning the terminal session.

C. Testbed configuration

The testbed was deployed at the *campus* of the University of São Paulo. A traffic-free street with $\approxeq 500$ m in length, as shown in Figure 2. The end points A and C indicate the places where the vehicle returns during the experiment, and the point B the RSU's positioning. To collect measurements, we use

TABLE II. WIRELESS ROUTER SPECIFICATIONS FOR ON-BOARD UNIT (OBU) AND ROAD-SIDE UNIT (RSU)

Device	**Description**
ALIX.3D2	PC-Engine Motherboard with 500Mhz CPU, 32Mb RAM and two USB ports
XtremeRange 5 XR5 802.11a	Ubiquiti XR5 Interface *Wireless* MiniPCI
Omni Antenna	Omnidirectional antenna with a gain of 5dBi and *dualband*: 2.4 and 5GHz
Redundant PoE	PoE (*Power Over Ethernet*) to plug the battery
Compact Flash	4GB of flash memory
Battery	$12V$ battery to independent power supply
GPS BU-353	GPS *GlogalSat SiRF Star* III
Magnetic foot	Magnetic foot to place the antenna

(a) (b)

Fig. 1. (a) Picture of the vehicle's roof with antenna and GPS receiver. (b) Picture of the RSU with antenna, router, battery and GPS on top of the rod.

a basic vehicle equipped with OBU as described in previous section. The RSU was equipped with same communications capacity of the vehicle's one and was fasten in the street edge with a steel rod.

Beaconing applications running on both OBUs and RSUs were responsible for collecting periodical beacon messages and saving them all into local log files that were used as dataset. During the experiment, the vehicle moved among points A, B and C. Beacons were transmitted and received by both RSU and OBU. More details about the testbed can be found in [11].

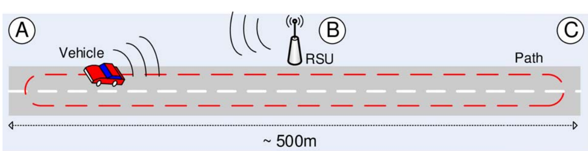


Fig. 2. Testbed scenario for data collect

TABLE III. WIRELESS COMMUNICATION CAPACITIES

Features	**Values**
Transmission power	23 dBm
Bandwidth	20 MHz
Frequency	5.8 GHz
Bit rate	6 $Mbps$
Channel	56
Wireless technology	IEEE 802.11a
Omnidirectional antenna	5dBi of gain

D. Measurements

During data collection, the vehicle completed nine laps in the testbed with variable speed, from $1\ m/s$ up to $14\ m/s$. Both, vehicle and RSU, were sending beacons at a frequency of $5\ Hz$. The logs files were recorded on the vehicle and in the RSU, with the following tuple: sender MAC address, receiver position, sender position, RSSI and vehicle speed.

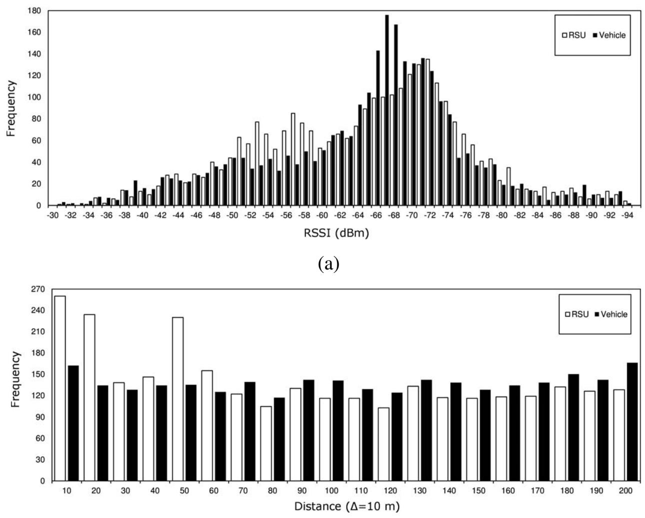
The datasets had to be filtered to remove inconsistent data portions of beacons received in the beginning and in the ending of the experiment. A disposable portion of data was collected when the vehicle should be in movement but it was temporarily stopped, e.g. when parking to initiate or interrupt the beaconing application or during the manoeuvre to return the vehicle. It is important to obtain an suitable amount of data for analysis of long distances, while the vehicular experiment is prone to high packet losses, hence, substantial lack of data. Thus, due the lack of data for longer distances and dropped portion of the log where the vehicle was not in movement, we had to pre-process the datasets by selecting tuples of beacons received in a maximum distance of $200\ m$. Regarding vehicular application requirements, the distance of $200\ m$ was enough to validate and evaluate the proposed architecture.

III. RSSI CHARACTERIZATION

We performed two well-known analysis to characterize RSSI. First, the distributions of the RSSI measurements and the amount of beacons per distance for both vehicle and RSU. Second, the average and standard deviation of RSSI as a function of the distances interval, $\triangle$.

Figure 3(a) shows the histogram of RSSI, the observation empirically suggests that the RSSI follows a normal distribution with mean between -70 to -72 dBm. Figure 3(b) shows the distribution of the amount of beacons collected as a function of the distance interval, $\triangle = 10\ m$. We observe that the samples have at least 100 beacons for each interval. The amount of RSSI beacons in the intervals 10, 20 and 50 m may have influenced the RSSI histogram from -50 to -59 dBm.

Figure 4 shows the average of RSSI for vehicle and RSU, which was computed using a distance interval $\triangle = 10\ m$. In general, the average was a reasonable parameter to represent the RSSI behaviour. As expected, it is possible to observe that the RSSI values tend to decrease when the distance increases. However, the means alternate in the intervals from 90 to 120 m and for distances greater than 170 m. If we observe standard deviations, they are greater for these intervals (see Figure 4). Another behaviour to notice is the increasing of RSSI as function of the distance in the short distances up to



(a)

(b)

Fig. 3. a) Frequency of RSSI for each 1 dBm and b) Frequency of beacons by distance interval ($\triangle = 10\ m$)

30 m. We believe this behaviour is consequence of the GPS position error. Since the distances were calculated based on the positions from GPS, for short distances this error impacts in the RSSI average.

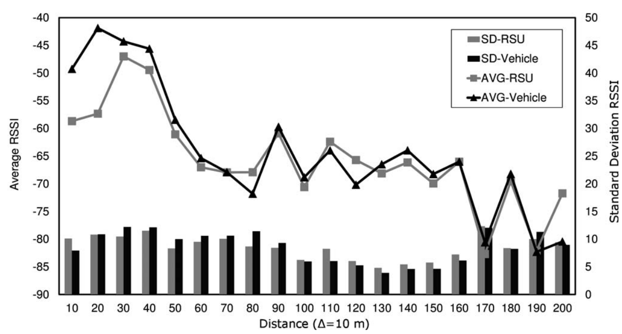


Fig. 4. Average and standard deviation of RSSI for vehicle and RSU

This analysis shows that is difficult to apply the path loss logarithmic equation directly on a single value of RSSI to compute the distance, due the fluctuation of the signal. For a good distance estimation with the path loss equation, the RSSI distribution should to be more uniform, also it needs adjust in the parameters of the equation model and computes the distance based on an average RSSI values. Therefore, this study exploits alternative techniques to estimate the distance based on the RSSI in the architecture proposed.

IV. PROBLEM STATEMENT

Let v be a vehicle in a vehicular network. During a travel, v needs to estimate the distance, $\bar{d}$, to the other vehicles or infrastructure r_j. Each r_j sends beacons at a frequency f. Assuming that each vehicle v has the capacity to store n instances $\hat{I}_n$, with RSSI information and the distance from r_j, which is computed from the beacons sent by r_j and received by v. If GPS fails, by applying a technique $\mathbb{k}$, v computes the estimated distance, $\bar{d}$, in relation to r_j, through an instance $\hat{I}^j$ generated from an amount, m, of beacons received from r_j.

Hypothesis: $\forall\ v$ that has stored n instances $\hat{I}_n$ is able to estimate the distance $\bar{d}$ in relation to r_j by the technique $\mathbb{k}$.

V. ARCHITECTURE

The proposed architecture is shown in Figure 5. This architecture has the advantages: *i)* fully distributed; *ii)* it is not mandatory all nodes of the vehicular network to implement the solution; *iii)* it is independent of the number neighbour vehicles; *iv)* it does not need line-of-sight to the r_j; and *v)* it is flexible to adapt to different traffic conditions. The main idea is to allow the vehicle, while traveling, store information from beaconing communication procedure. When a GPS fails, the vehicle uses the stored information to compute the distance to other vehicles or RSU. To this end, the architecture was designed with two procedures: *Database procedure* and *Estimate procedure*.

Database procedure is the phase which the vehicle, v, periodically processes and stores the information from different sources, such as: v's current geographic position obtained from GPS and RSSI from the wireless interface driver. Also, other information from the vehicle sensors could be useful to estimate the distance, e.g. temperature, air pollution, environment (city, tunnel, highway), etc, but the application of theses information are beyond the scope of this study. In the scenario of vehicular networks is reasonable to assume that vehicles send beacons at a frequency of 2 Hz. Thus, depending on the traffic density, the vehicle can receive dozens beacons per second. To handle this amount of information, in this phase we summarize the information and save in an instance, $\hat{I}$, which defined as set of parameters $\hat{I} = \{\bar{X},\ \bar{s},\ R,\ \triangle\}$, where: $\bar{X}$ is the average of RSSI; $\bar{s}$ is the standard deviation of RSSI; R is the amplitude of RSSI; and $\triangle$ is the distance interval. The parameters are calculated for each sample size, m, of beacons received in sequence that were sent by r_j.

Figure 6 shows the instances of $\hat{I}$. To better visualization of the instances, we plot $\bar{X}$ versus $\bar{s}$ from a subset with sample size, $m = 5$, and distance interval $\triangle = 50\ m$. The wide range of $\triangle$ is to make the instances more distinguishable to facilitate the visualization. Even with long distance interval, $\triangle = 50\ m$, it is not possible to completely separate the instances according to distance intervals. We noticed that there is an overlap of instances so that it is difficult to procedure distance estimation. For example, the intervals 100 $\vdash$ 150 m and 150 $\vdash$ 200 m are fused in some parts, with RSSI on average from -70 to -65 dBm and standard deviation from 0 to 5, including distances intervals from 100 to 200 m.

Estimate procedure is the phase which starts operating in the absence of geographical position, i.e. upon the GPS failure, to estimate distances to other vehicles or infrastructure, r_j. To infer the distance, the vehicle, v, computes an instance test, $\hat{I}^j$, based on the amount m beacons, that it has been continued receiving from r_j. The instance test is defined as $\hat{I}^j = \{\bar{X},\ \bar{s},\ R\}$ and, once computed $\hat{I}^j$, the vehicle v uses the technique $\mathbb{k}$ to compare its instances, $\hat{I}_{1..n}$, in the database in order to find the most similar instance to $\hat{I}^j$. Thus, v infers the distance to r_j by the distance, $\triangle$, saved in the $\hat{I}$.

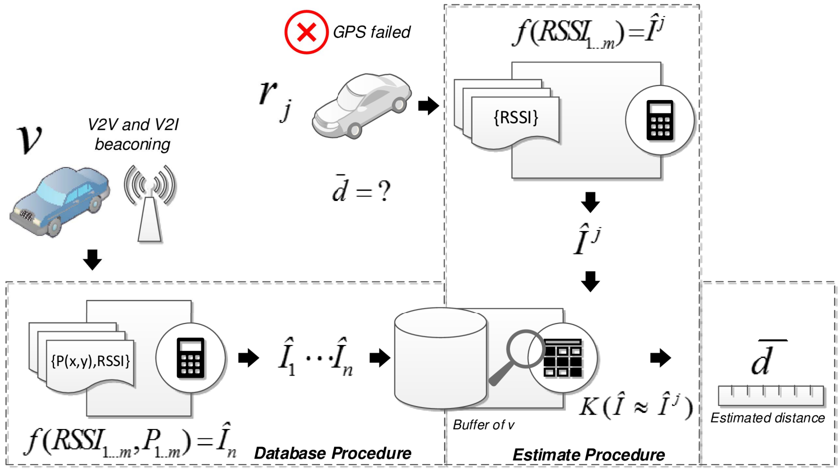


Fig. 5. Architecture to estimate distances in vehicular network using RSSI

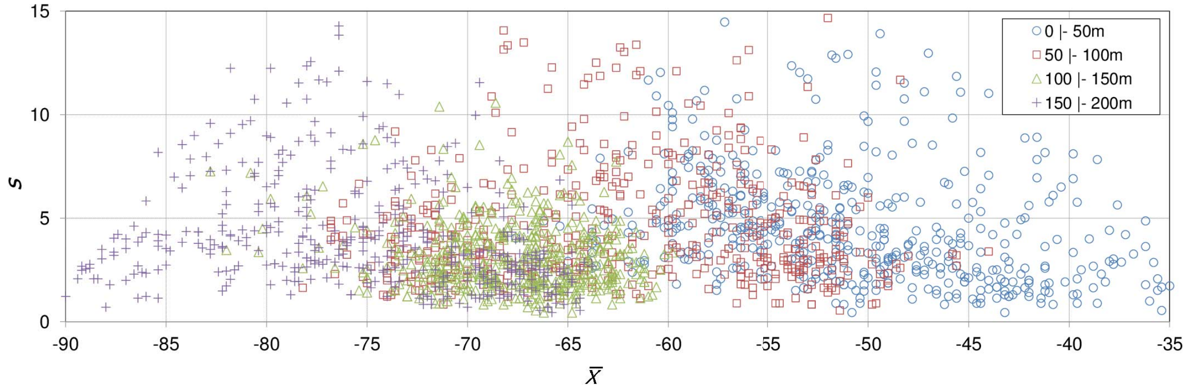


Fig. 6. Example of instances defined by $\hat{I} = \{\bar{X}, \bar{s}, \triangle\}$

This architecture can be extended to infer the v localization by using triangulation or multilateration techniques of distances of at least three different points of reference [12].

VI. Performance Evaluation

In this Section VI, we discuss the evaluation of the proposed architecture to compute the distance of the vehicle to infrastructure (from v to r_j) and the infrastructure to the vehicle (from r_j to v) from the dataset collected in our testbed. Subsection VI-A describes the methodology. Subsection VI-B shows a general performance evaluation of the most important cases, then we selected the parameters values for subsequent evaluations; Subsection VI-C discusses the impact of the parameter k; Subsection VI-D discusses the impact of the sample size m; Finally, Subsection VI-E shows the results for each distance interval.

A. Methodology

To evaluate the architecture and validate the hypothesis, we use the pre-processed data from the experiment in our testbed, as described in Section III.

We choose the k-nearest neighbors (k-NN) algorithm to compare the instances $\mathbb{k}(\hat{I}_{1..n}, \hat{I}^j)$ by regression. The metric used in the k-NN was the Euclidean distance defined in Equation 1, where x is a parameter in $\hat{I}^j = \{\bar{X}, \bar{s}, R\}$ and the number of parameters is $\rho = |\hat{I}^j|$.

$$\mathbb{k}(x_i, x_j) = \sqrt{\sum_{z=1}^{\rho}(x_{i,z} - x_{j,z})^2}. \tag{1}$$

Based on the characterization of the RSSI described in Section III, we discard the beacons less than 30 m in the log files of vehicle and RSU. Therefore, the evaluation is related to the interval 30 $\vdash$ 200 m. We sort each log by the distance and load it sequentially. For each m read beacon, we generated an instance, $\hat{I}$, until complete one instance per meter. The reminder of the beacons were used as instance test $\hat{I}^j$. For the overall permanence evaluation, we compare the k-NN technique versus free-space model.

B. General evaluation

For a performance overview, we initially used the parameters and values described in Table IV. We combine the parameters $\bar{X}$, $\bar{s}$ and, R. Each combination was chosen to evaluate the position measure (average $\bar{X}$) and dispersion measures (standard deviation $\bar{s}$ and amplitude R).

TABLE IV. PARAMETERS AND VALUES FOR A GENERAL EVALUATION

Parameter	Description	Value
$\triangle$	Distance interval	10 m
m	Sample size of the generated instances	10 beacons
k	Number of selected neighbouring instances	3 instances
n	Number of instances $\hat{I}$ in the v database	170 instances
$\aleph$	Number of instances test $\hat{I}^j$ evaluated	82 instances

Table V shows the results of each combination evaluation. The first column shows the acronym for facilitating the understanding and representation for the next evaluation. The second column are percentage of the hit, τ (%), that means a total of the hit for each combination. This is calculated by taking the real distance $\bar{d}$ of the v to r_j and compare with the estimated distance interval. The best τ is 27% to Xs-RSU, witch is the estimation of the RSU to vehicle using two parameters of $\hat{I}^j$, average $\bar{X}$ and standard deviation $\bar{s}$. The third column is the absolute mean error in meters, calculated by $\varepsilon = \frac{\sum_{e=1}^{\aleph} |\bar{d}_e - d_e^r|}{\aleph}$, where $\bar{d}$ is the estimated distance and d^r is the real distance. Again, the Xs-RSU has the best performance with a $\varepsilon = 14.21\ m$ of absolute mean error. Finally, the last column gathers parameters used in the k-NN Euclidean distance function, in which the evaluation was made with at most three parameters.

In this first evaluation, we notice that the increasing of parameters in the k-NN did not improve both mean error, ε, and hit, τ. In addition, the speed of the vehicle could affect the mean error, ε, and reduce the hit of τ, when comparing the performance the stand RSU (see Xs-RSU) with a moving vehicle (see Xs-Vehicle).

TABLE V. GENERAL EVALUATION OF THE PARAMETERS FOR THE VEHICLE AND RSU

Acronym	Hit τ (%)	Mean error ε (m)	Parameters of $\hat{I}^j$
Xs-Vehicle	20	16.41±3.13	$\bar{X}$ and $\bar{s}$
XR-Vehicle	18	16.83±3.42	$\bar{X}$ and R
XsR-Vehicle	17	16.69±3.38	$\bar{X}$, $\bar{s}$, and R
Xs-RSU	27	14.21±2.75	$\bar{X}$ and $\bar{s}$
XR-RSU	26	15.02±3.10	$\bar{X}$ and R
XsR-RSU	23	15.34±3.02	$\bar{X}$, $\bar{s}$, and R

Figure 7(a) is the Cumulative Distribution Function (CDF) of the absolute mean error for the distances estimates by Xs-Vehicle and Xs-RSU. Figures 7(b) and 7(c) show the error distribution for each distance interval, $\triangle = 10\ m$, of Xs-RSU and Xs-Vehicle, respectively. The results show that in both RSU and vehicle occur a distance estimation with high error in the interval $160 \vdash 180\ m$. Such an error is explained by the RSSI standard deviation (see Figure 4) for this interval.

C. Impact of k value

In the k-NN algorithm, the value of k should be according to the application performance. When $k > 1$, it means that the algorithm chooses the k most similar instances. To calculate the regression, we compute the average of distance values of the k similar instances.

Figure 8 shows the absolute mean error, ε, for Xs-RSU and Xs-Vehicle, applying different k values. The mean error has small variation for $k = 3$ considering RSU, but there are no significant reduction in error for vehicle and RSU. Therefore, for $k > 3$ does not impact on the error reduction.

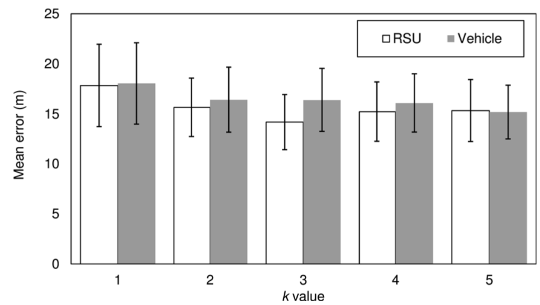


Fig. 8. Mean error ε for different k values

D. Impact of sample size m

The architecture uses the parameter of sample size, m, to generate the instances. For small, m, the vehicle, v, generates more instances, since in the vehicular network the nodes send beacons in frequency at least 2 Hz, e.g. when $m = 2$, one instance/second is generated. On the other hand, due to the high fluctuation of the signal, only two samples may not be enough to properly represent the range of distances. Therefore, it is necessary to adjust the m value according to the period of beacons and the precision of the distance estimation.

Figure 9 shows absolute mean error ε for increasing values of m from 6 to 14. The result shows that up to $m = 14$ there are small reductions in the error ε. The best performance is the reduction in the mean error when $m = 12$. Hence, sample size of $m = 12$ to generate an instance $\hat{I}$ implies that the reference r_j needs to send 12 beacons in sequence. This can cause overhead on the communication channel.

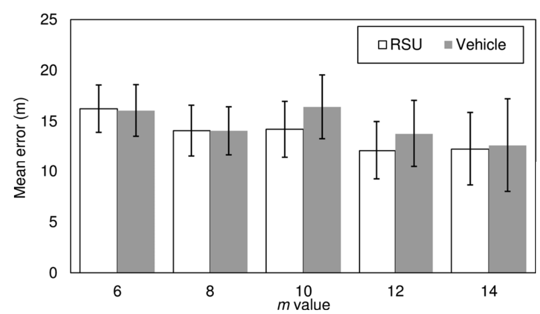
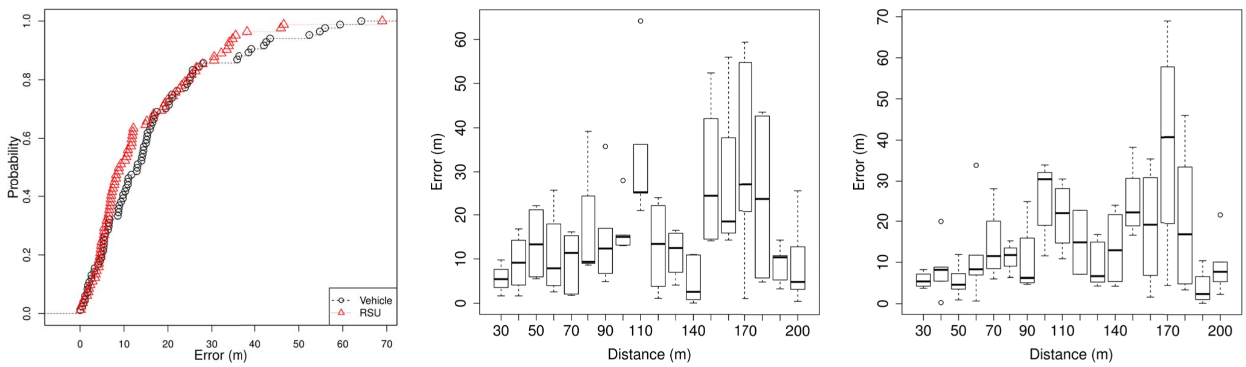


Fig. 9. Mean error ε for different m values

E. Overall performance

Traditionally, free-space path loss (FSPL) model is the straightforward way to forecast the received signal strength when the sender and receiver have a clear path between them, i.e. a line of sight. The received power in free space from a receiving antenna, which is separated from a transmitting antenna by a distance d (in km) and frequency f (in GHz), according to the Equation 2.

$$FSPL(dB) = 20\ log_{10}(d) + 20\ log_{10}(f) + 92.45 \quad (2)$$



(a) (b) (c)

Fig. 7. a) CDF of the mean error of Xs-RSU and Xs-Vehicle; b) Distribution of the error in meters for each distance interval, $\triangle = 10\ m$, of the vehicle; and c) Distribution of the error in meters for each distance interval, $\triangle = 10\ m$, of the RSU

To complete the evaluation, we compare the use of k-NN technique versus the well-know free-space model. First, we computed the free-space for the same communications parameters values using in the testbed, as shown in Table VI. Second, we compare the RSSI for the same distances used in k-NN as shown in Figure 10. RSSI free-space curve has lower attenuation compared with the RSSI average of the vehicle and RSU experimental data (see Figure 4 and Figure 10). The curves of the experimental RSSI average (AVG-RSU and AVG-vehicle logarithms), in the Figure 10, were calculated by a trendline tool available in spreadsheet softwares based on the logarithm behaviour. Third, we adjusted the free-space curve to approximate the real attenuation using a constant C in dBm to increase the original free-space attenuation ($FSPL + C$). The adjusted values curve are shown in the line "Free-space + C", for a $C = -13\ dBm$. Finally, to estimated the distances, for each instance $\hat{I^j}$ created with $m = 12$, we took the RSSI average and computed the distance by free-space adjusted model.

TABLE VI. COMMUNICATION PARAMETERS FOR FREE-SPACE.

Parameter	Value
Attenuation model	*Free-Space*
Gain of transmitting antenna(G_t)	5 dBi
Gain of receiver antenna (G_r)	5 dBi
Carrier frequency (f)	5.8 Ghz
Transmission Power(P_t)	23 dBm (200 mW)

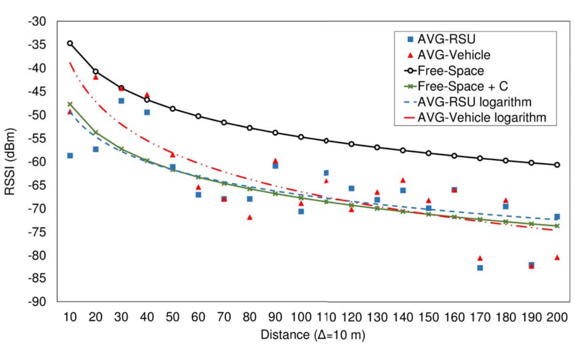
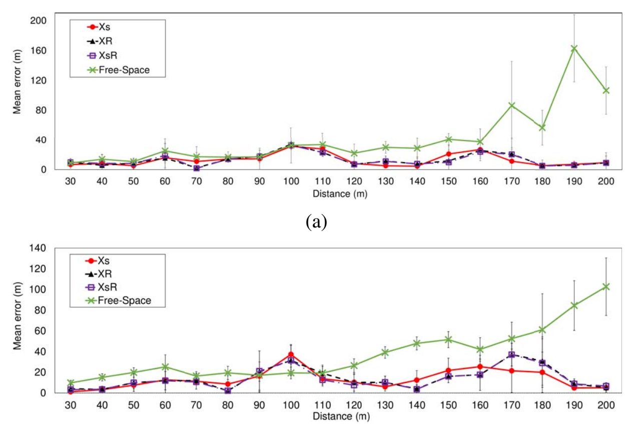


Fig. 10. Free-space attenuation versus RSSI real data experiment

Figures 11(a) and 11(b) show the absolute mean error ε for each distance interval, $\triangle = 10\ m$, considering the combinations of the parameters $\bar{X}$, $\bar{s}$, R and free-space model. The results of k-NN technique show that there are two intervals with error greater than the average, $80 \vdash 110\ m$ and $150 \vdash 180\ m$. We believe that it is caused by the RSSI distribution (see Figure 3(a)), where the instances of the two intervals are overlapped (see Figure 6) as consequence of the instances $\hat{I}$ representation, based on the statistical position and dispersion measures.



(a)

(b)

Fig. 11. a) Mean error ε for the RSU, and b) Mean error ε for the vehicle

For free-space results, due the instances $\hat{I}$ stores the average of RSSI, the representation by instances applied to the free-space model reaches a reasonable error, ε, until the distance $\triangle = 120\ m$, after the error starts a high degradation of the accuracy, even the RSSI being an average, it means that the free-space model did not fit well to the real data for distances higher than 120 m. Therefore, we note that the architecture is feasible and the use of instances based in statistical measures can reach reasonable results. In this context, the k-NN shows better results than the well-known free-space model. We believe that the error can minimize for specific applications, such as localization based on RSSI, where it's possible to take into account at least three different sources/references of signal.

VII. CONCLUSION

This paper addresses the problem of measuring distances between vehicle and RSU with only RSSI values to assist positioning applications. We propose an architecture to estimate distances that tolerates GPS device failures in vehicular networks. The solution uses the RSSI obtained from beaconing communication to calculate the distance intervals across the instances. The architecture is evaluated using dataset collected in real environment experiment. The results demonstrate the feasibility of the architecture and that the absolute mean error ε in the estimated distance is around $16\ m$ and $14\ m$ for vehicle and RSU, respectively. Moreover, the hit rate τ reaches 20% and 27% for vehicle and RSU, respectively, which means the total hit of the distance interval measurements. When comparing the free-space model with k-NN, the results of k-NN are better than free-space for distances higher than 120 m.

For a future work, the architecture will be extended to calculate a relative position of the vehicle based on the distances measure by RSSI.

ACKNOWLEDGMENT

The authors would like to thank the supporting organizations for funding this work: The agency CAPES and grant #2015/18264-0, São Paulo Research Foundation (FAPESP).

REFERENCES

[1] Aviles, J.V.M.; Prades, R.M., "Pattern Recognition Comparative Analysis Applied to Fingerprint Indoor Mobile Sensors Localization". In: Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on, pp.730–736, 2010.

[2] Chiang, J. T.; Haas, J. J.; Hu, Y.-C. "Secure and precise location verification using distance bounding and simultaneous multilateration". In: Proceedings of the Second ACM Conference on Wireless Network Security, ACM Conference on, (WiSec09), pp. 181–192, 2009.

[3] Malandrino, F.; Casetti, C.; Chiasserini, C.F.; Fiore, M.; Yokoyama, R.S.; Borgiattino, C., "A-VIP: Anonymous verification and inference of positions in vehicular networks". In: INFOCOM, Proceedings IEEE, 14-19 April, pp.105–109, 2013.

[4] Souza, A. M.; Maia, G.; Yokoyama, R. S.; Loureiro, A. A. F.; Villas, L. A. "Minimizing Traffic Jams in Urban Centers Using Vehicular Ad Hoc Networks". In: New Technologies, Mobility and Security (NTMS), 7th International Conference on, 27–29 July, 2015.

[5] Tiedeke, J.; Schabel, P.; Rille, E. "Vehicle distance sensor using a segmented IR laser beam". In: Proceedings of the 40th Vehicular Technology Conference, IEEE Conference on, pp. 107–112, 1990.

[6] Vandenberghe, W.; Moerman, I.; Demeester, P. "Approximation of the IEEE 802.11p standard using commercial off-the-shelf IEEE 802.11a hardware". In: 11th International Conference on ITS Telecommunications (ITST), pp.21–26, 2011.

[7] Widmann, D.; Balac, K.; Taddeo, A.; Prevostini, M.; Puiatti, A. "Characterization of in-tunnel distance measurements for vehicle localization". In: Proceedings of the Wireless Communications and Networking Conference, IEEE Conference on, (WCNC13), pp. 2311–2316, 2013.

[8] Ying, Y.; Yuhui, Z. "Technique of measuring leading vehicle distance based on digital image processing theory". In: Proceedings of the Intelligent Computation Technology and Automation, IEEE Conference on, (ICICTA10), pp. 674-677, 2010

[9] Yokoyama, R.S.; Kimura, B.Y.L.; Jaimes, L.M.S.; Moreira, E.D.S. "A Beaconing-Based Opportunistic Service Discovery Protocol for Vehicular Networks". In: Advanced Information Networking and Applications Workshops (WAINA), 28th International Conference on, pp.498–503, 13-16 May, 2014.

[10] Yokoyama, R.S.; Kimura, B.Y.L.; Moreira, E.D.S. "An architecture for secure positioning in a UAV swarm using RSSI-based distance estimation". SIGAPP Appl. Comput. Rev. 14, v.2, pp.36–44, June 2014.

[11] Yokoyama, R.S.; Kimura, B.Y.L.; Moreira, E.D.S. "V-Beacon: Uma plataforma experimental para Redes Veiculares Sem Fio". In: III Simpósio Brasileiro de Engenharia de Sistemas Computacionais (SBESC), 2013.

[12] Yokoyama, R.S.; Kimura, B.Y.L.; Moreira, E.D.S. "Secure positioning in a UAV swarm using on-board stereo cameras". In: Proceedings of the 29th Annual ACM Symposium on Applied Computing (SAC'14), pp.769–774, 2014.