A Decentralized Traffic Monitoring System Based on Vehicle-to-Infrastructure Communications

José Geraldo R. Júnior[*†], Miguel Elias M. Campista[*], and Luís Henrique M. K. Costa[*]

[*] Grupo de Teleinformática e Automação, UFRJ - COPPE/PEE - DEL/Poli, Rio de Janeiro – RJ – Brazil
Email: {jgrjunior, miguel, luish}@gta.ufrj.br

[†] CEFET-MG - Campus III - Leopoldina - MG - Brazil
Email: {jgrjunior}@leopoldina.cefetmg.br

***Abstract*—Current proposals of automated traffic monitoring systems for highways require high investment in terms of installation and maintenance, especially because these systems rely on a central element which is responsible for inferring and disseminating the traffic conditions. This paper presents DTraMS (Decentralized Traffic Monitoring System), a system that monitors and disseminates traffic conditions using a decentralized infrastructure. On board units and road side units do not necessarily need to be interconnected, nor connected to a central computer. These elements exchange information to update their tables that describe the traffic conditions of each route section. Vehicles serve as data mules that disseminate the information, whereas road side units store and combine the information received from them. To validate DTraMS, a prototype was implemented using IEEE 802.11b/g and experiments were performed on the campus of Federal University of Rio de Janeiro. The comparison of the results obtained with DTraMS and data obtained with a GPS shows that DTraMS is accurate in detecting both the position of the vehicle and on estimating the road condition.**

I. Introduction

According to the European DG TREN (Directorate-General for Energy and Transport), the number of cars increases about 3 million per year [1] in Europe. Only in the Brazilian State of Rio de Janeiro, this number grows by 250,000 every year. In 2013, the state accounts already for more than 5 million vehicles on the roads. The large number of vehicles in circulation has an enormous impact on the amount of traffic jams experienced by drivers. In this scenario, transportation departments all around the world are looking for new traffic monitoring solutions that are hopefully less expensive, more reliable, and more automated [2].

Traffic monitoring is frequently performed by visual inspection, e.g., by means of video cameras strategically positioned [3], [4]. These systems depend on human control and are inefficient since they can neither react nor foresee potential circulation problems. Another commonplace option is the use of sensors installed along the roads [5], [6]. These sensors, however, may have a high cost. For instance, each inductive loop detector, including hardware and controllers, can cost several thousands of dollars (around US$8,000) [2]. Another important characteristic of the present scenario is the incredible dissemination of smartphones. As a consequence, recently proposed automated traffic monitoring systems combine the use of GPS (Global Positioning System) [7], [8] with 3G [9], [10] or IEEE 802.11 networks [9], [11]–[13] to determine the vehicle density on the road. In most of these systems, the gathered information is sent to a central unit for compilation and publishing. From the smartphone viewpoint, the shortcoming of combining GPS with 3G/4G is battery consumption, three times larger than only using IEEE 802.11 [13], [14]. An additional issue of the 3G/4G technology, besides the cost to the final user, is the limited maximum rate imposed to users to avoid network congestion [15].

In this paper, we propose DTraMS, a low-cost Decentralized Traffic Monitoring System to provide information regarding vehicle speed as an alternative to the current traffic monitoring systems. The basic idea is to rely on user cooperation to keep the system economically viable and scalable. Users only need a mobile device equipped with an IEEE 802.11 interface to participate in the system. Using only the information available on IEEE 802.11 beacon frames, which is anyway periodically sent by access points (APs) [16], DTraMS infers the traffic conditions, collaboratively. In DTraMS, each On-Board Unit (OBU) is responsible of inferring its speed on each road segment and to send the obtained measurements to Road Side Units (RSUs). Upon receiving such information, each RSU updates a local database, called an SCT (Segment Condition Table), which contains speed information about each road segment. It is worth mentioning that the entire process does not require connection between RSUs. This way, DTraMS can be used even on roads where there is no communication infrastructure connecting RSUs. This aspect, however, leads to a challenge concerning synchronization. Since there is no centralized element, it is not possible to guarantee the synchronization between the all of the device clocks. To circumvent this shortcoming, we use a control mechanism similar to the lease control of DHCP (Dynamic Host Configuration Protocol) [17], where the lifetime is in seconds and each device decreases the time to zero, based on the local clock. Although DTraMS uses IEEE 802.11 to define the vehicle location, it is possibile to use GPS. Therefore, since the GPS is not needed, we save battery which is frequently an issue for smartphone applications.

We have implemented a DTraMS prototype using IEEE 802.11 networks. The experiments conducted in the Campus of the Federal University of Rio de Janeiro (UFRJ), Brazil, show that it is possible to monitor the traffic conditions based on the received signal strength indicator (RSSI). We have evaluated the accuracy of the results by comparing them to GPS data, showing that the obtained values are quite similar. The results also attest that with a signal power as low as -60 dBm, we have a discrepancy of less than 3 meters when comparing OBU and RSU position (previous experiments present similar results even in scenarios with high number of vehicles and IEEE 802.11 networks [11].).

This paper is organized as follows. Section II details the traffic monitoring system architecture, including its algorithms, and the prototype implemented using IEEE 802.11 b/g. Section III analyzes the experimental results. Finally, Section IV concludes this work and investigates possible future directions.

II. DTraMS

Figure 1 presents the elements of the DTraMS architecture: the mobile on-board units inside vehicles and static road-side units. The OBU may be any portable device running the client application inside a vehicle on the road. It could be a laptop, a tablet, a smartphone or any other device that has a network interface that supports the IEEE 802.11 b/g. An RSU, on the other hand, is composed of an access point inside an hermetic box [18]. In the example scenario of Figure 1, RSUs are installed at bus stops. We consider that users inside vehicles have IEEE 802.11 devices, the road has traffic flowing in both directions, and that the distances between RSUs are known. As there is no connection between RSUs, vehicles fulfill the role of communication links, allowing each RSU to have information about traffic conditions on the road.

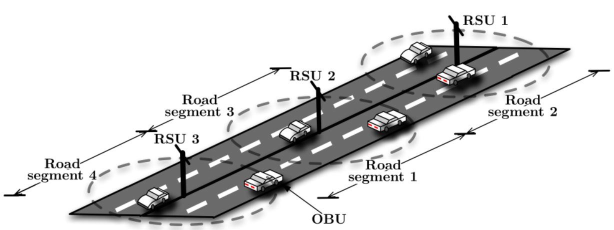

Fig. 1: DTraMS architecture.

We assume define a road segment as the portion of road between two consecutive RSUs (Figure 1). As a consequence, the number of road segments is proportional to the number of RSUs (N_{RSU}). Because each direction of the road is considered a different road segment, the number of road segments, N_{RS}, is given by:

$$N_{RS} = (N_{RSU} - 1) \times N_D, \quad (1)$$

where N_D is the number of directions. In this proposal, N_D is always a multiple of two.

In our architecture, each unit maintains an important data structure, called a Segment Condition Tables (SCT), wihich is used to exchange information between OBUs and RSUs. Figure 2 illustrates the structure of SCTs which are composed of entries representing each road segment. These entries, i.e., each line on SCT, have three fields: `Road Segment`, road segment `Condition`, and `TTL` (Time To Live). The `Road segment` is the unique identifier of each road segment, which is used for table comparison. The `Condition` field represents the current average speed in the road segment, and the `TTL` represents the lifetime of each table entry. The TTL has basically three functions: it defines the lifetime of each entry, assigns higher weight to more recent entries, and minimizes the lack of synchronization between clocks. Section II-B presents more details about the TTL utilization in DTraMS.

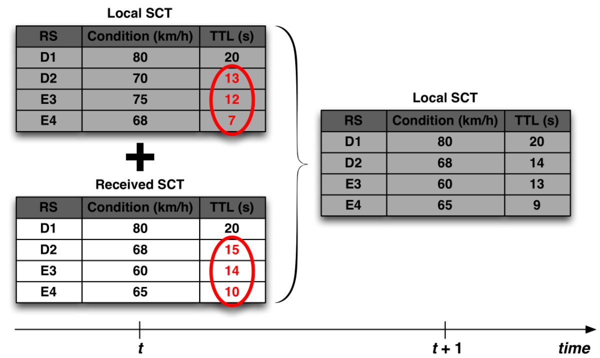

Fig. 2: Update of the local SCT, in OBU, after the reception of a new information from RSU.

In DTraMS, the RSSI value is used to define the moment when the OBU crossed an RSU. Since the received power value varies as the vehicle moves, we consider that the OBU is closest to an RSU when it observes the maximum RSSI value from it. The value of RSSI is simply obtained from the beacon frames of IEEE 802.11 networks. Beacons are periodically sent by IEEE 802.11 access points, by default in the standard, every 100 ms [16]. In total, we use three informations obtained from the beacons: the ESSID (Extended Service Set Identification), the BSSID (Basic Service Set Identification) of the RSU, and the power level of the received signal. The ESSID is the network identifier in the IEEE 802.11 standard. In DTraMS, it serves also as the road identifier. The BSSID, on the other hand, is the MAC address of the access point, which corresponds to the RSU in DTraMS. This information can be captured even if the OBU is not yet associated to the RSU, because it is gathered from the 802.11 beacons. It suffices to use the network interface in monitor mode before the OBU associates with the RSU.

A. Algorithms of DTraMS

In DTraMS, the OBU is responsible to compute the mean speed in the road segment, update the local SCT, detect the best moment to send the local SCT to the closest RSU, and send the local SCT to this RSU. Submitting this information, the RSU can at least update the entries of its SCT concerning previous road segments, since the new TTL values are higher.

Analyzing Figure 1, when the OBU reaches the end of the road segment 1, it updates and sends the local SCT to RSU 2, where at least the information from road segments 1 and 4 will be latest. Although the vehicle has not passed by the segment 4, it has received the segment condition from RSU 1.

1) On-Board Unit: Algorithm 1 presents the routine executed by an OBU. As shown in the algorithm, each OBU searches a known ESSID within received beacon frames (lines

5 and 41) and, upon finding one, it stores the RSSI values and tries to associate to the RSU (lines 6 to 9). After the association process, the OBU receives the SCT from the RSU and compares it with its local SCT, entry per entry (lines 10 to 19) (initially, the TTL value is zero). It is worth mentioning that the number of entries is equal to number of road segments. When the TTL of the received SCT is higher than the TTL of the local SCT, the information about the road segment is updated. Figure 2 illustrates the comparison, where the local table, at instant $t+1$, is updated after comparing the tables at instant t.

The next step is to define the moment when the vehicle passes by an RSU. The OBU compares two values, the highest and the last RSSI value received. When the maximum RSSI received is 10 dBm higher than the last RSSI received, we consider that the vehicle has already passed by the RSU and is getting more distant. The interval of 10 dBm is based on results obtained in previous experiments [11]. The OBU uses the time interval between the two last RSUs as estimated to calculate the mean speed on the road segment. The OBU updates the local SCT and sends the table to the RSU. These steps are represented in the lines 20 to 32 of the Algorithm 1.

Finally, after disconnecting (line 30, of the Algorithm 1), it is important to avoid a new OBU association to the previous RSU which it has just passed by (lines 31 and 32).

2) Road Side Unit: Algorithm 2 presents the routines executed by an RSU. As shown in Algorithm 2, each RSU announces its current SCT to the OBU which is completing the association process (lines 3 and 4). This way, the RSU shares the traffic conditions about the road with nearby OBUs, in both directions. When the RSU receives an SCT from an OBU, it also needs to compare the received information with its local information (lines 4 to 8). If the received TTL for a road segment is higher, it updates its entries (lines 9 to 11). Nevertheless, in most cases, we may have several vehicles sending data simultaneously. Since vehicles may present different mean speeds according to each lane condition, the RSU calculates the harmonic mean to infer the current condition of the road segment (lines 13 to 16). Using Equation 2, the system assigns a higher weight to the most recent information.

$$HM_{RS}(t) = \frac{2}{\frac{1}{v_i} + \frac{1}{HM_{RS}(t-1)}}, \quad (2)$$

where v_i is the OBU mean speed and $HM_{RS}(t-1)$ is the previous harmonic mean value on the road segment.

To identify in which road segment the OBU is, or define the vehicle direction, we use the last and the present RSUs. This verification is done using a local XML file, in the format of Listing 1. It is possible to check in the list the extension of the road segment and that the road segment is composed of two RSUs (lines 4, 8, and 16). The local file also stores information about each RSU, in this example: the ESSID (lines 7 and 15), MAC addresses, and geographic coordinates (lines 10, 11, 18, and 19). We consider that the client application knows the road where the vehicle is traveling. This is a input value of the client application.

Figure 3 shows the steps of the communication between OBU and RSU as a diagram. DHCP operations fall into four basic steps: IP discovery, IP lease offer, IP request, and IP lease acknowledgment. The required time on each stage is presented in our experimental results in Section III.

B. TTL Mechanism

In DTraMS, we do not assume synchronization between clocks inside OBUs and RSUs since the scenario is partially disconnected. Nevertheless, we assume that each device is responsible for decrementing its local TTL based on local time. Thus, the TTL value is decremented every second. The lack of synchronization is minimized and restricted to the time needed for the SCT exchange between the RSU and the OBU.

Listing 1. Road segment definition file.

```
1  <root>
2      <path id='ID'>
3          &lt;segment&gt;1&lt;/segment&gt;
4          <distance>500</distance>
5          <from>
6              <essid>"Avenida Brasil Digital"</essid>
7              <macAddress>00:26:5a:a6:13:67</macAddress>
8              <location>
9                      <latitude>-22.8649888333</latitude>
10                     <longitude>-43.2149178333</longitude>
11             </location>
12         </from>
13         <to>
14             <essid>essid</essid>
15             <macAddress>00:26:5a:a6:13:70</macAddress>
16             <location>
17                     <latitude>-22.8638633333</latitude>
18                     <longitude>-43.2189613333</longitude>
19             </location>
20         </to>
21     </path>
22 </root>
```

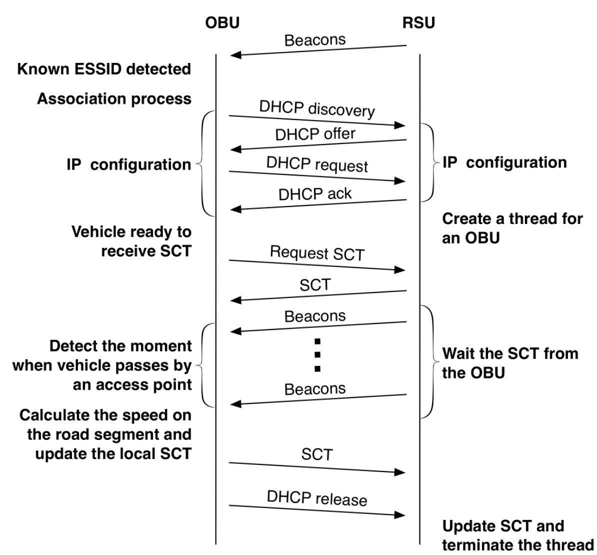

Fig. 3: DTraMS diagram.

The maximum TTL value depends on the road characteristics, such as the number of lanes, length, and speed limit. On the one hand, a maximum TTL too small may not allow the update of the road conditions in distant road segments. The slower the road speed limit, the worse is this effect because

Algorithm 1 Algorithm executed by an OBU.

Input: RSSI, ESSID, and BSSID of the RSU.
Output: The instant when the OBU crosses the RSU.

$num_{RS} = ((num_{RSU} - 1) * num_D)$;
$actualPower \leftarrow 0$;
$maxPower \leftarrow 0$;
while true **do**
  Search for known ESSID; //monitor mode
  if ESSID is known **then**
    $actualAP \leftarrow MAC_{AP}$;
    Store $actualPower$ in a ordered vector;
    Try to associate;
    if It is associated **then**
      Receive SCT from the RSU;
      $numLine \leftarrow 1$;
      while $num_{Line} < num_{RS}$ **do**
        if $TTL_{recent_{Line_x}} < TTL_{received_{Line_x}}$ **then**
          $TTL_{recent_{Line_x}} \leftarrow TTL_{received_{Line_x}}$;
          $Condition_{recent_{Line_x}} \leftarrow Condition_{received_{Line_x}}$;
        end if
        $num_{Line} \leftarrow num_{Line} + 1$;
      end while
      while Associated **do**
        if $|maxPower| > |receivedPower|$ **then**
          $|maxPower| \leftarrow |receivedPower|$
          Store $actualPower$ in an ordered vector;
        else
          if $|receivedPower| - |maxPower| >= 10$ **then**
            Store the instant of OBU passed by RSU;
            $v(V_x, RS_y) \leftarrow \frac{ext(RS_y)}{t_x(RSU_y) - t_x(RSU_{y-1})}$;
            Update local SCT;
            Send the SCT updated to associated RSU;
            Disconnect;
            $previousAP \leftarrow recentAP$;
            $recentAP \leftarrow \{\}$;
          end if
          Store $recentPower$ in vector;
        end if
      end while
    else
      Try to associate with the RSU;
    end if
  else
    Search by known ESSID on received beacon frames;
  end if
end while

vehicles take longer to reach distant road segments. On the other hand, an overestimated maximum TTL can result in outdated information, especially desceibing farther segments. To compute the TTL value, we use the following equation:

$$TTL > p(i) \times \sum_{i=1}^{n} t_{RS_i}, \tag{3}$$

where i is the current road segment, $p(i)$ is the probability of traffic jam occurrence in that road segment, n is the number of road segments, and t_{RS_i} is the mean time to pass by the current road segment. The probability of traffic jam occurrence ($p(i)$) is a input value, based, i.e., on historical data.

Algorithm 2 Algorithm executed by an RSU.

Input: The instant when the OBU has crossed the RSU.
Output: Harmonic mean of road segments.

$num_R S = ((num_{RSU} - 1) * num_D)$;
while true **do**
  Receive association request;
  Send the current local SCT plus the network settings;
  Wait for an updated SCT;
  Receive the updated SCT;
  $numLine \leftarrow 1$;
  while $num_{Line} < num_{RS}$ **do** //Compare received SCT with local SCT;
    if $TTL_{Recent_{Line_x}} = 0$ **then**
      $TTL_{Recent_{Line_x}} \leftarrow TTL_{Received_{Line_x}}$;
      $Condition_{Recent_{Line_x}} \leftarrow Condition_{Received_{Line_x}}$;
    else
      if $TTL_{Recent_{Line_x}} < TTL_{Received_{Line_x}}$ **then**
        $HM_{RS}(t) = \frac{2}{\frac{1}{v_i} + \frac{1}{HM_{RS}(t-1)}}$;
        $TTL_{Recent_{Line_x}} \leftarrow TTL_{Received_{Line_x}}$;
        $Condition_{Recent_{Line_x}} \leftarrow HM_{RS}(t)$;
      end if
    end if
    $num_{Line} \leftarrow num_{Line} + 1$;
  end while
end while

III. Field Experiments and Analysis

In the experiments, we use a real scenario set up in the campus of the Federal University of Rio de Janeiro, Brazil. The RSUs are installed inside hermetic boxes and power-supplied by 12 V batteries and a 5 V voltage regulator, as shown in Figures 4 and 5. The client application runs on a Sony Vaio laptop with a processor Intel I5-3210m, 6 GB of RAM, 640 GB of hard disk, and an IEEE 802.11 network interface. For comparison purposes, we use a GPS model `u-Blox 5`, which gives the OBU position four times per second. The client application was implemented using Python.

In our experiments, we use two RSUs and one OBU. Figures 4 and 5 present the road segments. We had other vehicles travelling at the same time in different lanes. The OBU measures the association time (starting when OBU receives the first beacon from the RSU), the time required for IP configuration, the time to execute the exchange of tables, and the moment of disconnection. It is worh mentioning that our experimentation scenario was prone to intereferences from other wireless networks. We have detected 11 IEEE 802.11 b/g networks in channel 9 or 11. We use the channel 11 in the OBU.

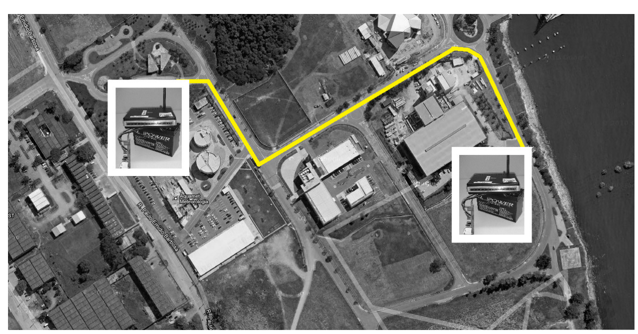

Fig. 4: Road segment 1 of the scenario - UFRJ.

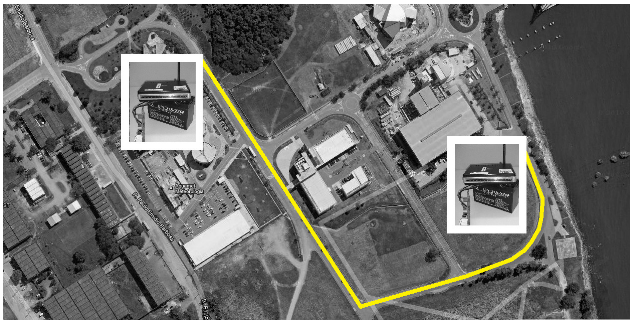

Fig. 5: Road segment 2 of the scenario - UFRJ.

Figure 6 shows an experiment where the OBU crosses an RSU. The figure depicts the time intervals corresponding to each stage of the DTraMS algorithm. The moment when each stage finishes is represented. The mean speed of the vehicle is about 40 km/h. The entire process takes about 17 seconds (from 15:05 to 15:22). The OBU always disconnects after passing by an RSU, when the RSSI value is 10 dBm lower than the maximum RSSI measured.

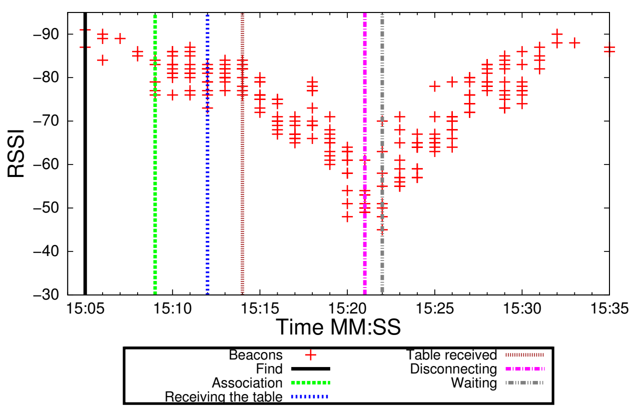

Fig. 6: Steps of the DTraMS: Beacons, RSSI behavior of the received beacon from RSU; Find, instant when the OBU detects a known ESSID; Association, end of association process; Receiving the table and Table received, interval between request of the SCT and response; Disconnecting, instant when OBU disconnect to RSU; and Waiting, waiting to next RSU.

Figure 7 presents the DTraMS behaviour in 8 rounds. Each round represents the moment when the vehicle reaches the end of a road segment. The time interval between the moment that the OBU detects a known ESSID and when it associates to the RSU varies from 6 to 11 seconds. It took 3 to 9 seconds to receive the network settings from the RSU plus the time needed to receive and send the SCT. The mean time to receive the SCT is 2 seconds. Experiments using fixed IP address in the OBU have reduced the time needed to finish the network configuration from 3-9 seconds to less than 1 second. The association and the reception of network settings are the steps that take longer. In this case, we can think of at least two options to reduce the time needed for the algorithm execution when using IEEE 802.11 b/g. The first is the the utilization of a predefined IP address in OBUs. The second option would be to make the first access point as the sole responsible for distributing IP addresses. In the second option, the initial step would only happen when the vehicle enters the road.

Another option is to consider IEEE 802.11 p [19], the 802.11 variant which is part of the WAVE (Wireless Access in Vehicular Environment) protocol suite. In the IEEE 802.11p standard, a station in “WAVE mode” can transmit and receive data frames without establishing a BSS (Basic Service Set). Consequently, OBUs do not need to wait for the association and authentication procedures before exchanging data. Stations with IEEE 802.11 p use the wildcard BSSID in the frames header. Using 802.11 p, the association time would be reduced. In the WAVE protocol family, the WAVE Short Message Protocol (WSMP) is an alternative to TCP/UDP and IPv6. WSMP allows applications to have direct control over physical layer characteristics, such as channel and transmission power used to send messages. The sending application also provides the MAC address of the target device, although there is the possibility of using the broadcast address. Even if IEEE 802.11 p is specific for vehicular networks, we have used IEEE 802.11 b/g for two main reasons. First one is the possibility of evaluating the proposal in real scenarios since only a few 802.11 p wireless cards are curently comercially available. The second reason is because IEEE 802.11 b/g/n, besides its much larger availability, is present on a much larger variety of equipment, including smartphones, tablets, and laptops, and there is currently no reason for those type of devices to be equipped with the 802.11 p variant.

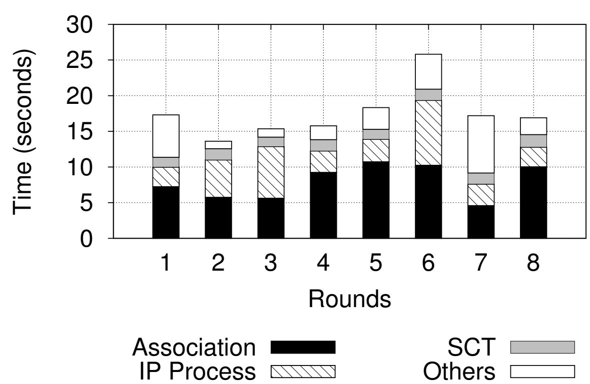

Fig. 7: Histogram - 8 rounds.

To analyze the accuracy of DTraMS, we compare the OBU and RSU positions in the moment when the received signal power is higher. The results show that with a received signal power as low as −60 dBm we have a discrepancy of less than 3 meters. Previous experiments present similar results even in scenarios with high number of vehicles and 802.11 networks [11]. According to Boukerche *et al.* [20], the error of location systems varies between 10 and 30 m. Therefore, our result can be considered satisfactory since errors are minimized by predictable movements of client nodes.

Figure 8 presents the traffic condition inferred on each road segment by both DTraMS and GPS. To generate a single condition using the GPS, we have calculated the harmonic mean of all values obtained in the road segment. By default in the standard, used GPS define the OBU location every 0.25 seconds. For comparison purposes, we have varied this interval, via client application. The obtained results are very close, showing that the efficiency of DTraMS is quite similar to the best case, even using IEEE 80211 b/g.

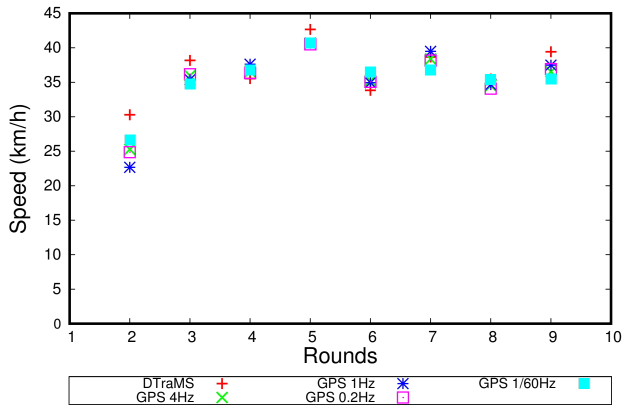

Fig. 8: Comparison between DTraMS and GPS results.

IV. Conclusions

This paper has presented DTraMS, a collaborative system for distributed monitoring of traffic using vehicular networks. In DTraMS, road-side units (RSUs) do not necessarily need to be connected to a control central, since vehicles fulfill the role of communication links, allowing each RSU to have information about traffic conditions on the road. Thus, DTraMS is well suited for highways without cellular coverage or energy infrastructure.

Experiments were carried out in a real scenario with very similar results to those obtained by a high-precision GPS. DTraMS is accurate in detecting both the position of the vehicle, with discrepancy of less than 3 meters, and on estimating the road condition, focus of this paper.

As future work we aim at simulating the DTraMS in larger scenarios using NS-3 [21], aiming at analyzing the amount of traffic generated by the system, its scalability with respect to the number of cars, as well as the minimum number of cars needed for the system to provide the traffic conditions.

Acknowledgments

This work was partially funded by CAPES, COFECUB, FAPERJ, CNPq, and FINEP/FUNTTEL.

References

[1] Directorate-General fo Energy and Transport, "European energy and transport. trends to 2030," European Commision, Tech. Rep., 2004. [Online]. Available: http://ec.europa.eu/dgs/energy/transport/figures/trends/2030/

[2] Y. Gongjun, S. Olariu, and D. Popescu, "Notice: An architecture for the notification of traffic incidents," *Intelligent Transportation Systems Magazine, IEEE*, vol. 4, no. 4, pp. 6–16, 2012.

[3] J. Hubaux, S. Capkun, and J. Luo, "The security and privacy of smart vehicles," *Security Privacy, IEEE*, vol. 2, no. 3, pp. 49–55, May-June.

[4] M. Raya, P. Papadimitratos, and J.-P. Hubaux, "Securing vehicular communications," *Wireless Communications, IEEE*, vol. 13, no. 5, pp. 8–15, Oct. 2006.

[5] S. Y. Cheung, S. Coleri, B. Dundar, S. Ganesh, C.-W. Tan, and P. Varaiya, "Traffic Measurement and Vehicle Classification with a Single Magnetic Sensor," 2004, Institute of Transportation Studies, Research Reports, Working Papers, Proceedings. [Online]. Available: http://EconPapers.repec.org/RePEc:cdl:itsrrp:qt2gv111tv

[6] K. Kwong, R. Kavaler, R. Rajagopal, and P. Varaiya, "Arterial travel time estimation based on vehicle re-identification using wireless magnetic sensors," *Transportation Research Part C: Emerging Technologies*, vol. 17, no. 6, pp. 586 – 606, 2009.

[7] Q.-J. Kong, Q. Zhao, C. Wei, and Y. Liu, "Efficient traffic state estimation for large-scale urban road networks," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, no. 1, pp. 398–407, 2013.

[8] 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*, ser. SenSys '08. New York, NY, USA: ACM, 2008, pp. 323–336.

[9] Google, "Google Maps," 2011, accessed in August 2013. [Online]. Available: Availableinhttp://maps.google.com/support

[10] D. Valerio, A. D'Alconzo, F. Ricciato, and W. Wiedermann, "Exploiting Cellular Networks for Road Traffic Estimation: A Survey and a Research Roadmap," in *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*, April 2009, pp. 1 –5.

[11] J. G. Ribeiro Júnior, M. E. M. Campista, and L. H. M. K. Costa, "Opportunistic System for Collaborative Traffic Monitoring Using Existing IEEE 802.11 Networks," in *Intelligent Vehicular Networking: V2V/V2I Communications and Applications - IEEE International Conference on Communications - ICC*, Ottawa, Canada, Jun. 2012, pp. 7294 –7298.

[12] J. A. Sanguesa, M. Fogue, P. Garrido, F. J. Martinez, J.-C. Cano, C. T. Calafate, and P. Manzoni, "An infrastructureless approach to estimate vehicular density in urban environments," *Sensors*, vol. 13, no. 2, pp. 2399–2418, 2013.

[13] 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*, ser. SenSys'09. New York, NY, USA: ACM, 2009, pp. 85–98.

[14] A. Carroll and G. Heiser, "An Analysis of Power Consumption in a Smartphone," in *Proceedings of the 2010 USENIX conference on USENIX annual technical conference*, ser. USENIXATC'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 21–21.

[15] A. Balasubramanian, R. Mahajan, and A. Venkataramani, "Augmenting Mobile 3G using WiFi," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, ser. MobiSys '10. New York, NY, USA: ACM, 2010, pp. 209–222.

[16] "IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *ANSI/IEEE Std 802.11, 1999*, pp. i –513, 2003.

[17] IETF RFC 2131, "Dynamic Host Configuration Protocol," Mar. 1997.

[18] J. G. Ribeiro Júnior, L. H. M. K. Costa, M. E. M. Campista, I. M. Moraes, R. S. Alves, R. S. Couto, F. O. B. Silva, L. G. Valverde, M. L. D. Lanza, B. C. V. Camilo, and M. D. Amorim, "GT-ReBUS: Network Access on Bus of Universities," Available in http://www.gta.ufrj.br/gt-rebus, Tech. Rep., 2011, acessed in August 2013.

[19] Y. Du, L. Zhang, Y. Feng, Z. Ren, and Z. Wang, "Performance analysis and enhancement of ieee 802.11p/1609 protocol family in vehicular environments," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, 2010, pp. 1085–1090.

[20] A. Boukerche, H. A. B. F. Oliveira, E. F. Nakamura, and A. A. F. Loureiro, "Vehicular ad hoc Networks: A New Challenge for Localization-Based Systems," *Comput. Commun.*, vol. 31, pp. 2838–2849, July 2008.

[21] NS-3 Consortium, "Network simulator - v3," 2008, acessed in July 2013. [Online]. Available: http://www.nsnam.org/