TrafficView: A Driver Assistant Device for Traffic Monitoring based on Car-to-Car Communication

Sasan Dashtinezhad, Tamer Nadeem
Department of Computer Science
University of Maryland
College Park, MD 20742
{*sasan,nadeem*}*@cs.umd.edu*

Bogdan Dorohonceanu, Cristian Borcea,
Porlin Kang, Liviu Iftode
Department of Computer Science
Rutgers University
Piscataway, NJ 08854
{*dbogdan,borcea,kangp,iftode*}*@cs.rutgers.edu*

***Abstract*— TrafficView is a device that can be embedded in the next generation of vehicles to provide the drivers with a real-time view of the road traffic far beyond what they can physically see. The vehicles equipped with TrafficView devices disseminate traffic information using short-range wireless communication. The main benefits of disseminating traffic information in a vehicle-to-vehicle fashion are scalability and ease of deployment. This paper describes the TrafficView prototype and presents preliminary experimental results for this prototype.**

I. Introduction

The emergence of standardized low cost wireless connectivity such as 802.11, miniaturization of computing devices, and availability of Global Positioning System (GPS) for civilian use open up opportunities for many new outdoor distributed applications. One important area which will benefit from this new trend is intelligent transportation systems [1].

To realize an intelligent transportation system, a common platform for inter-vehicle communication is needed. This platform should support safe driving, dynamic route scheduling, emergency message dissemination, and traffic condition monitoring. Existing solutions for reporting accidents or traffic conditions rely on certain infrastructures, such as traffic monitors located along the road reporting data to a central database, or cellular wireless communication between vehicles and a monitoring center. Users can query the aggregated information from a central database via cellular networks. The problem with such solutions is that they require expensive infrastructures installed on every road the system is going to be used. Additionally, they are not scalable due to their centralized design.

TrafficView [2] provides a lower cost, simpler deployment, and more scalable alternative to existing solutions. TrafficView is a device that can be embedded in vehicles to provide the drivers with a real-time view of the road traffic far beyond what they can physically see. The vehicles equipped with TrafficView form an ad hoc vehicle-to-vehicle network to exchange traffic information. Figure 1 shows an example of traffic information displayed to a driver by a TrafficView device.

Unlike infrastructure-based networks (e.g., cellular networks), TrafficView devices construct an ad hoc network on-the-fly, which changes dynamically to reflect the current traffic situation. More importantly, the short-range wireless networks work even on roads outside of the coverage of infrastructure-based networks. Furthermore, TrafficView is inherently scalable due to its completely decentralized design.

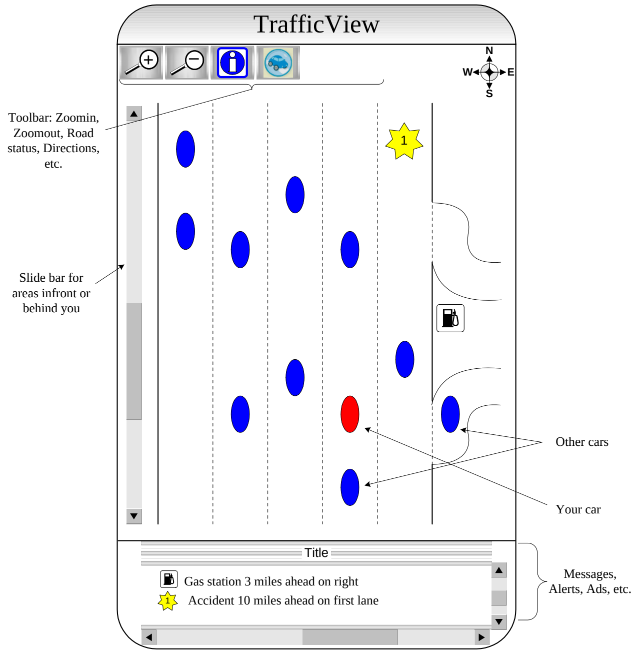

Fig. 1. Example of Traffic Information Displayed by TrafficView

The rest of this paper presents the TrafficView prototype and is organized as follows. Section II describes our prototype, and Section III presents the details of the navigation system incorporated in the prototype. Preliminary experimental results for the prototype are shown in Section IV. We discuss related work in Section V and conclude in Section VI.

II. TrafficView Prototype

Each TrafficView device installed in a vehicle is composed of three components: an embedded computer with display, a GPS receiver, and a short-range wireless network interface (*802.11b*). Optionally, an *on-board diagnostics system (OBD) interface* [3] can be used to acquire mechanical and electrical data from sensors installed in vehicles. The GPS receiver provides the location, speed, current time, and direction of the vehicle. Each vehicle equipped with TrafficView gathers and

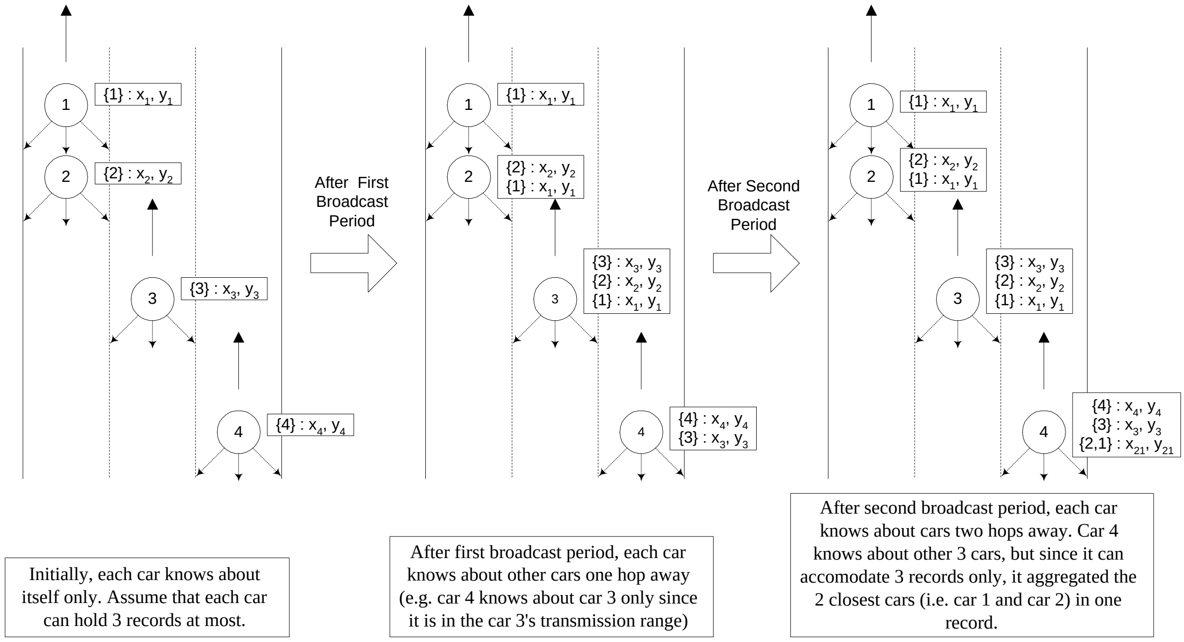

Fig. 2. TrafficView Scenario

broadcasts information about other vehicles in a peer-to-peer fashion using the short-range wireless interface. The display shows a map of the road ahead of the vehicle, annotated with dynamic and real time information exchanged over the ad hoc wireless network (as illustrated in Figure 1).

Figure 3 presents the TrafficView software architecture. Each vehicle stores records about other vehicles in its local *validated* dataset. Local GPS readings are periodically generated and adjusted through the navigation module before storing them. When a record is received through a broadcast message, it is stored in the *non-validated* dataset, since it might contain outdated or conflicting information. After these records are examined for validity, they are merged with the validated dataset. Periodically, the system displays data from the validated dataset.

Each vehicle periodically broadcasts *aggregated* data from the validated dataset (Figure 2). Given the trade-off between the ability to present the driver with as much information as possible and the limited wireless bandwidth, TrafficView uses semantic data aggregation mechanisms to reduce the size of the data sent over the network. For instance, if a vehicle has two records about two vehicles which are close to each other and are moving with relatively the same speed, an aggregation algorithm can replace those two records with one record representing both vehicles (see Figure 2). We have designed and evaluated through simulations scalable aggregation algorithms for TrafficView [2].

Our prototype is implemented in Java over Linux. As a testbed, we have used HP iPAQs equipped with 802.11 cards for wireless communications.

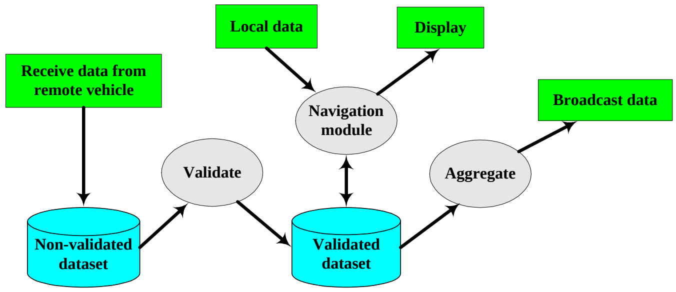

Fig. 3. TrafficView Software Architecture

III. Navigation System

In TrafficView, the relative positions and directions of other vehicles with respect to the current vehicle as well as the distances between vehicles are needed. To solve this problem, the road on which each vehicle runs, the closest mapped point on that road, and the direction of motion with respect to that road are needed at any given time. A mapped point is a node in a graph that represents the roads and their connections in a geographical area. The graph, along with other information, is part of a map of that area. Once this information is known, the distances and moving directions among the vehicles can be computed. We have developed a software module that processes the output from a Garmin GPS device [4] using the Garmin Simple Text Output Protocol (GSTOP) [5]. An example of a GSTOP reading is given in Figure 4.

Since the number of meters per latitude or longitude degree differs dramatically with the degree, all the computations involving distance are done using conversion tables from latitude and longitude degrees to meters [6]. The corresponding meter

@031222181226N4031717W07428034G014+00043E012.3N0031U0001

Fig. 4. Example of a GSTOP character line: start of GPS reading line: @, date(YY/MM/DD): 03/12/22, time(hh:mm:ss): 18:12:26, location: 40 degrees and 31.717 minutes North and 74 degrees and 28.034 minutes West; reading obtained using 3D GPS positioning method, horizontal position error: 14m, altitude above the sea level: 43m, velocities: 12.3 m/s Eastward; 3.1 m/s Northward; and 0.01 m/s Upward.

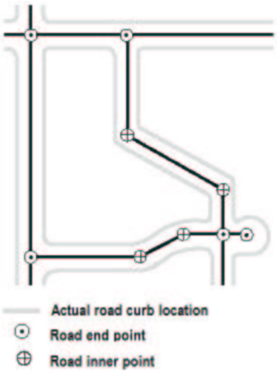

Fig. 5. Road representation in Tiger Line Files of type RT1 and RT2

values for degree values between two consecutive latitude or longitude degrees are computed using interpolation.

To create the road graph, Record Type 1 (RT1) and Record Type 2 (RT2) files [7] from the data files offered by the US Census Bureau through the 2000 Tiger Line database [8] are used. Each road is recorded as a set of reference points that represent extremities of segments in a chain. RT1 files contain information about the roads, such as name, type, direction, start, and end points whereas RT2 files contain information about intermediary reference points on the roads described in RT1 files. There is a set of RT files for every county in each state of the US [9]. Figure 5 shows an example of a generalized block for Tiger Line map recorded in RT1 and RT2 files (only the road graph is shown).

Identifying the location on the map based on a sampled GPS position involves identifying the closest reference map points (recorded in RT files). Therefore, we would like to have a mechanism that clusters the map points; this mechanism should be minimal space-wise and search-wise due to the limited memory and computing resources of an embedded system.

We have designed a mechanism based on *simple Peano keys* [7], which can be used to collapse a 2D coordinate system into a 1D coordinate system. Given a pair of two numbers (e.g., 1111 and 2222), a simple Peano key can be obtained by interleaving their digits (i.e., 12121212). To generate a simple Peano key for a point on the map, we interleave the digits of the numbers that give the longitude and latitude. If we create simple Peano keys for each point on a map and put them in a sorted array in lexicographic order, the adjacent keys represent map points that are close to each other on the map.

Figure 6 presents an example of our mechanism. The last key from the sorted array used in comparison by the binary

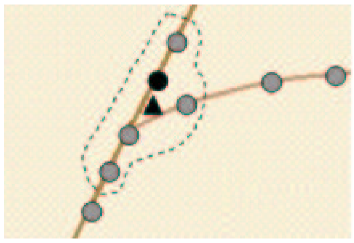

Fig. 6. Identifying the road closest to a certain GPS-provided position. The dots are map points generated from Tiger Line information. The triangle represent the GPS position. The dots in the enclosure are the map points that have the closest simple Peano keys to the simple Peano key of the GPS position for *r=2*. The closest map point to the GPS position is colored black, and its position determines the closest road

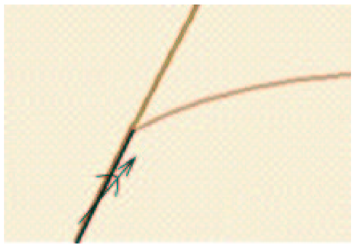

Fig. 7. Matching a GPS segment (bounded by two successive GPS location readings) to a road segment by computing the angle between the two segments

search is the closest Peano key and corresponds to one of the very close reference points. To improve the probability of choosing the closest reference point (i.e., situated at the minimum distance in space to the GPS position), we consider all the Peano keys in a *2r* interval around the closest Peano key (*r* keys before the closest key, and *r* keys after that).

Identifying the closest road does not guarantee that we have identified correctly the road on which the vehicle is running. Let us assume that a vehicle approaches an intersection where it may change its road. Deciding near (before, in, and after) the intersection on which road the vehicle is running is done in two steps. In the first step, the software module tries to identify the closest road which has approximately the same direction to the direction on which the vehicle is running. If two successively identified map points are on the same road, the software module computes the angle between that segment and the segment bounded by the corresponding two GPS sampled points, as shown in Figure 7 (where the road segment is shown as a thick segment, and the segment bounded by the two GPS sampled points is shown as an arrow). If the angle is less than 15 degrees, the vehicle is considered to be on, or very close to the road that contains the two reference map points. In the second step, the software module uses a smoothing veto technique similar with a short-term memory to decide the road. The road that appears most frequently in the last *m* identified roads is chosen to be the current road on which the vehicle is running.

Two problems may still lead to an incorrect identification of a road. First, the GPS measuring accuracy is between 3 and 30 meters, and therefore, a vehicle can be anywhere inside

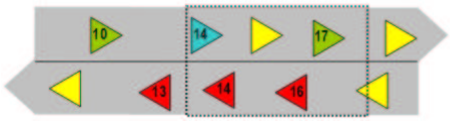

Fig. 8. Determining the direction of motion for vehicles

a circle with a radius between 3 and 30 meters. Second, the distance between two reference map points given by the Tiger Lines files can be as large as 200 meters. To drastically reduce road identification errors due to these factors, we divide each Tiger Line road segment in sub-segments creating equally distant reference points on it through interpolation such that any of the segments that form a road have length less than a predetermined value *d*.

To determine the direction of motion for vehicles, we use the following solution. For each road, the road points are stored in an array. Once two successive road points are identified, the direction in which the vehicle is moving on that road can be computed by looking at the direction in which the indices of the points in the array are decreasing or increasing. Therefore, for a set of vehicles in the TrafficView network, we can tell what vehicles are moving in one direction and what vehicles are moving in the opposite direction (Figure 8).

IV. Experimental Results

In this section, we present the evaluation of the performance of our prototype using real GPS traces obtained on a highway. We also present the evaluation of the accuracy of our navigation system.

For our first set of experiments, we have acquired 8 GPS traces by driving vehicles on a highway and recording time, latitude, longitude, and speed. The GPS traces are collected by driving on highway road of 10939m length with an average speed of about 15m/s. The cars were moving in a row with an average distance between each consecutive cars of 200m. This distance allows each car to hear the broadcast messages from the car in front of it and the one behind it, assuming the nominal wireless transmission range of 250m for the 802.11b standard[1]. Using these traces, our navigation module computed the distance to a reference point on the highway and output that along with the time and speed. We fed these outputs, as movement patterns for eight vehicles, in the TrafficView emulator of our prototype.

We measured the performance of the prototype in terms of visibility and accuracy achieved by our *ratio-based* aggregation algorithm [2] versus simple data propagation (i.e., without aggregation). The idea of the ratio-base algorithm is to divide the road in front of each vehicle into regions and perform more aggregation for fartehr away regions. Regions are defined by two sets of input parameters of the algorithm: *aggregation ratios* and *portion values*. For each region, the aggregation ratio determines how much aggregation is performed for the records about the vehicles in that region, and the corresponding portion value specifies how much of the remaining space in the broadcast packet is assigned to the region.

[1]In practice, we found out that the wireless transmission range is less than 250m. However, using external antennas, we can restore this transmission range.

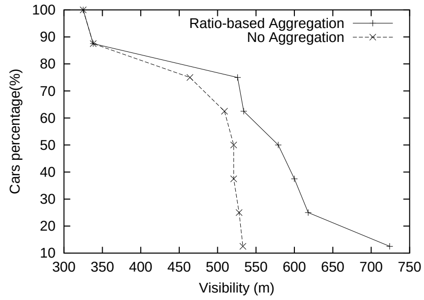

Fig. 9. Vehicle visibility

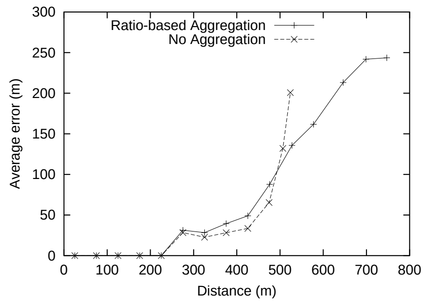

Fig. 10. Average position estimation error

Although our experiments used a small number of vehicles, we have found that the effect of the ratio-based aggregation is still significant compared to the non-aggregation case. Figure 9 shows the maximum vehicle visibility along the road. A point (d, p) on this graph indicates that $p\%$ of the vehicles have had a visibility of d meters or more. For instance, all the cars have maximum visibility of at least 300m ahead, whereas about 25% of the cars have visibility of at least 525m ahead. This percentage increases for the aggregation case, where about 75% of the cars have a visibility for more than 525m.

Figure 10 shows the accuracy, which is the average error in estimating the position of vehicles in front of each vehicle. As we can see, the accuracy of the aggregation mechanism is slightly worse than the non-aggregation case for cars within 500m ahead, while it outperforms the non-aggregation case for cars beyond 500m. This is because the cars in the non-aggregation case have a limited visibility, and most of the them have no information or non updated information about cars that are at least 500m away because of the small size of the broadcast packets we use.

Our second set of experiments quantify the accuracy of the navigation system. Table I shows the average accuracy of road identification, computed over 4 sample routes of 1441, 2628, 1582, and 1562 GPS readings in Middlesex county (34689 roads) of New Jersey using the tuning parameters $r = 5$ and $m = 10$ (defined in the previous section). The reference

d (m)	Map Points	Point Count (%)	Average Accuracy (%)
5	1578948	100	100
10	807215	51	97
15	550338	34	94
20	422032	26	92
25	344268	21	89

TABLE I

ROAD IDENTIFICATION ACCURACY USING VARIABLE MAXIMUM ALLOWED LENGTH *d* FOR ROAD SUBSEGMENTS.

map points were created only once during the initialization of the software module. The table shows that for a short length of road sub-segments (*d*=5), we achieve 100% identification accuracy. The accuracy decreases only slightly as this length increases from 5m to 25m, although the number of reference map points (columns 2 and 3) decreases sharply.

V. RELATED WORK

The idea of using short-range Inter-Vehicle communication for traffic safety applications has been exploited by several research groups. In [10], a wireless traffic light system is presented, where information about current light status, location of intersection, and a reference point are broadcasted periodically. In [11], the authors present a collision warning system that exchanges beacon messages in a peer-to-peer fashion. To the best of our knowledge, TrafficView is the first system which provides a dynamic view of traffic flow to a driver, based on GPS data and short-range wireless network connectivity.

The work presented in [12] describes an Inter-Vehicle Communication system (IVC) with Vehicle-Roadside Communication (VRC), where both moving vehicles and base stations can be peers in the system. TrafficView could use VRC to query traffic conditions in farther-away regions, which are not visible through our aggregation schemes. The CarNet [13] project focuses on providing IP connectivity to radio nodes in vehicles with the help of a grid location service [14].

In [15], the authors examine resource discovery using an *opportunistic* approach in inter-vehicle ad-hoc networks in an urban area, where moving vehicles communicate with each other via short-range wireless transmissions. Their work could be incorporate into the validation module of TrafficView, where we can handle the vehicles as resources.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented our experience with building and evaluating an initial prototype of TrafficView. The main goal of this system is to provide drivers with information about traffic and road conditions far beyond what they can physically see. To achieve this goal, vehicles equipped with TrafficView devices gather and disseminate traffic information among vehicles on the road. The information displayed by these devices allows drivers to be aware of the traffic condition ahead, which helps driving in situations like foggy weather, or finding an optimal route in a trip several miles long.

Currently, we are pursuing two different directions. The first one is to evaluate our prototype on the roads with a larger number of cars. The second one involves finding solutions for privacy and security concerns.

Privacy is an important issue in such a system. Different privacy levels should be available from which the drivers can select. One level of privacy could be to completely hide any information about the vehicle, while it continues to participate in relaying other vehicles' information. Another level is to allow other vehicles to gain information about the current vehicle without being able to identify it.

Security and trust are two other important issues in such a system. A fraudulent vehicle could disseminate information about non-existent vehicles, or broadcast bogus information about existing vehicles. We plan to investigate different mechanisms that prevent such situations and identify fraudulent vehicles.

VII. ACKNOWLEDGMENT

This work is supported in part by National Science Foundation under ANI-0121416.

REFERENCES

[1] "Intelligent Transportation Systems, U.S. Department of Transporation," http://www.its.dot.gov.

[2] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode, "TrafficView: A Scalable Traffic Monitoring System," in *Proceedings of the 2004 IEEE International Conference on Mobile Data Management (MDM 2004)*, Berkeley, GA, January 2004.

[3] "On-Board Diagnostic System," http://www.obdii.com/.

[4] "Garmin devices," http://www.Garmin.com.

[5] "Garmin, Simple Text Output Protocol," http://www.Garmin.com/support/text_out.html.

[6] S. Humerfelt, the Earth according to WGS 84. http://home.online.no/ sigurdhu/Grid_1deg.htm.

[7] "US Census Bureau, Tiger Line Technical Documentation," http://www.census.gov/geo/www/tiger/rd_2ktiger/tgrrd2k.pdf.

[8] "US Census Bureau, Topologically Integrated Geographic Encoding and Referencing system (Tiger Line)," http://www.census.gov/geo/www/tiger.

[9] "US Census Bureau, 2002 Tiger/Line Files by state," http://www.census.gov/geo/www/tiger/tiger2002/tgr2002.html.

[10] Q. Huang and R. Miller, "The design of reliable protocols for wireless traffic signal systems," Department of Computer Science and Engineering, Washington University, Tech. Rep., 2003.

[11] R. Miller and Q. Huang, "An Adaptive Peer-to-Peer Collision Warning System," in *IEEE Vehicular Technology Conference (VTC)*, Birmingham, AL, May 2002.

[12] I. Chisalita and N. Shahmehri, "A Peer-to-Peer Approach to Vehicular Communication for the Support of Traffic Safety Applications," in *Proceedings of the 5th IEEE International Conference on Intelligent Transportation Systems*, Singapore, Sept 2002, pp. 336–341.

[13] R. Morris, J. Jannotti, F. Kaashoek, J. Li, and D. Decouto, "CarNet: A Scalable Ad Hoc Wireless Network System," in *Proceedings of the 9th ACM SIGOPS European Workshop.* Kolding, Denmark: ACM Press, New York, NY, September 2000, pp. 61–65.

[14] J. Li, J. Jannotti, D. S. J. D. Couto, D. R. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, Boston, MA, Aug 2000, pp. 120–130.

[15] A. Ouksel, O. Wolfson, and B. Xu, "Opportunistic Resource Exchange in Inter-vehicle Ad Hoc Networks," in *Proceedings of the 2004 IEEE International Conference on Mobile Data Management (MDM 2004)*, Berkeley, GA, January 2004.