

TrafficView: Traffic Data Dissemination using Car-to-Car Communication*

Tamer Nadeem^a, Sasan Dashtinezhad^a, Chunyuan Liao^a

{nadeem,sasan,liaomay}@cs.umd.edu

^aDepartment of Computer Science, University of Maryland, College Park, MD, USA

^bDepartment of Computer Science Rutgers University, Brunswick, NJ, USA

Liviu Iftode^b

iftode@cs.rutgers.edu

Vehicles are part of people's life in modern society, into which more and more high-tech devices are integrated, and a common platform for inter-vehicle communication is necessary to realize an intelligent transportation system supporting safe driving, dynamic route scheduling, emergency message dissemination, and traffic condition monitoring. TrafficView, which is a part of the e-Road project, defines a framework to disseminate and gather information about the vehicles on the road. With such a system, vehicle's driver will be provided with road traffic information that helps driving in situations as foggy weather, or finding an optimal route in a trip several miles long. This paper describes the design and implementation of TrafficView and the different mechanisms used in the system.

I. Introduction

Vehicles are part of people's life in modern society, into which more and more high-tech devices are integrated. Most of the current research focuses on the functionalities of individual vehicles, and less attention has been paid to the cooperation among vehicles and road facilities, which forms the transportation system. Moreover, a common platform for inter-vehicle communication is necessary to realize an intelligent transportation system supporting safe driving, dynamic route scheduling, emergency message dissemination, traffic condition monitoring, etc.

The e-Road project is an attempt to achieve the aforementioned goals by providing a scalable infrastructure for inter-vehicle communication. Specifically, the e-Road project is aimed at building a system consists of: 1) *Real-time message dissemination platform* to be used in sending messages about traffic condition monitoring, road condition, accident report, road-side e-advertisements, etc., 2) *Information query platform* that enables vehicles to query for information about specific objects or places such as road condition at Exit 11, and 3) *Reliable information exchange protocol* to the connection-oriented applications such as music downloading, back-seat passenger games, or connection to the Internet.

In this paper, we present *TrafficView*, which is a part of the e-Road project. TrafficView defines a framework to disseminate and gather information about the vehicles

*This paper is an extended version of the paper "TrafficView: A Scalable Traffic Monitoring System" that appeared in "2004 IEEE International Conference on Mobile Data Management (MDM'04)". This work is supported in part by National Science Foundation under ANI-0121416.

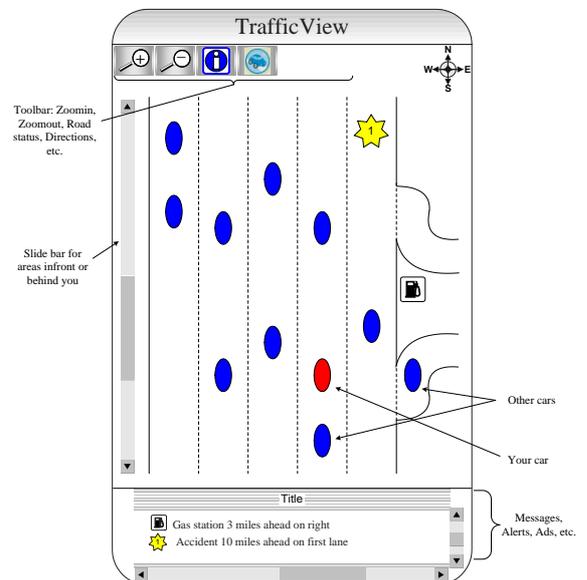


Figure 1: Example of Traffic Information Displayed by TrafficView

on the road. Using such a system, a vehicle driver will be aware of the road traffic, which helps driving in situations like foggy weather or finding an optimal route in a trip several miles long.

A GPS receiver shows a static view of the map, whereas TrafficView provides the driver with a dynamic view of the road traffic, and therefore complements the GPS receiver. When integrated with the traditional digital map system, TrafficView would be able to provide the functionality of real-time automatic route scheduling. Moreover, in such a platform, other applications such as accident alert, and road-side e-advertisement can be easily implemented. Figure 1 shows an example of traffic information displayed to a driver by TrafficView device. This paper describes our experience in developing the

TrafficView system. Throughout our experimentation, we performed a detailed study of different information dissemination techniques under various road density and vehicle mobility conditions.

The rest of the paper is organized as follows: The next section summarize the related work, and the description of the problem is given in Section III. In Section IV and Section V we describe the design of TrafficView and the mechanisms used in the system. The System performance is studied in Section VI. Finally we present our conclusions and future work in Section VII.

II. Related Work

The research in Inter-Vehicle-Communication has emerged in the past couple of years; mainly because it is a good experimental platform for Mobile Ad Hoc Networks (MANETs), and has a great market potential [8]. In addition to the similarities to MANETs such as short radio transmission range, low bandwidth, omnidirectional broadcast (at most times) and low storage capacity, inter-vehicle communication has its unique characteristics and challenges as well:

- *Rapid changes in link topology.* Because of the relative movement of the vehicles, the connectivity between vehicles is always changing. For example, if vehicles' speed is 60mph (25m/s), and the wireless transmission range is 250m, the connectivity between two vehicles could last for at most $500/25 = 20$ sec.
- *Frequently disconnected network.* In low vehicle density case, gaps between vehicles might be several miles, far beyond the transmission range of wireless networks. In turn, the disconnection time could be minutes. Such situation is common due to the fast movement of vehicles and high dynamic traffics.
- *Data compression/aggregation.* Wireless networks have a limited available bandwidth. In order to build a scalable system, data compression/aggregation mechanisms are required to save the bandwidth.
- *Prediction of vehicle's positions.* Vehicles run along pre-built roads, which remain unchanged over years. Therefore, given the average speed, current position, and road trajectory of a specific vehicle, the future position of that vehicle can be predicted.
- *Energy is not an issue.* Nodes, in sensor networks, are battery-powered and it is not easy to replace the battery after deployment. Hence, many efforts have been made to conserve energy in sensor networks. On the other hand, in a vehicle network, the vehicle itself can be used as a source of electric power, and therefore, energy is not a big issue.

Several major automobile manufactures and universities have begun to investigate in this field; GM research center in CMU [7], BMW Research Labs [16] and Ford Research Labs [11], Rice University [17][13], and Harvard University [4] are a few to name. CarNet [12] project focuses on how the radio nodes in the vehicles get IP connectivity with the help of Grid [9]. In [14], a wireless traffic light system is presented. At the intersection, a static control unit periodically broadcasts the current light status, location of intersection, and a reference point, using which the vehicles approaching the intersection can check their relative position and make a decision accordingly. They also designed collision warning system [11] in which peer-to-peer beacon message exchange is used.

An architecture of the vehicular communication is described in [5]. It integrates inter-vehicle communication (IVC) with Vehicle-Roadside Communication (VRC), where both moving vehicles and base stations can be peers in the system. The peers are organized into Peer Spaces for message exchange, in which flooding is the main method of delivery. Authors in [13] examine the feasibility of short range communication between fast moving vehicles using Bluetooth, and a mobile test-bed RUSH has been established in [17], composed of the fixed base station and mobile nodes on shuttle buses.

Two delivery modes known as pessimistic and optimistic forwarding are compared in disconnected vehicle networks in [4]. The experiment shows that the average delay in optimistic delivery is better. The authors of [3] propose a "wait-and-resend" scheme where a mobile node can cache the message for a while before new neighbors enter its transmission range, and [10] proposes an algorithm to dynamically modify the trajectories of the intermediate nodes to approach next available nodes, for relaying the message to the destination.

III. Problem Description

Given a set of moving vehicles on the road, the goal is to exchange information about the position and speed of those vehicles among them to enable each individual vehicle to view and assess traffic and road conditions in front of it. As the vehicles move along the road, they might enter the transmission range of some vehicles, and exit that of others. Figure 2 (a) shows an example of a road with four lanes, on which four vehicles are moving. Two main mechanisms could be used to achieve this goal: *flooding* and *diffusion*. In the flooding mechanism, each individual vehicle periodically broadcasts (pushes) information about itself. Whenever a vehicle receives a broadcast message, it stores it and *immediately* forwards it by rebroadcast the message. Obviously, this method is

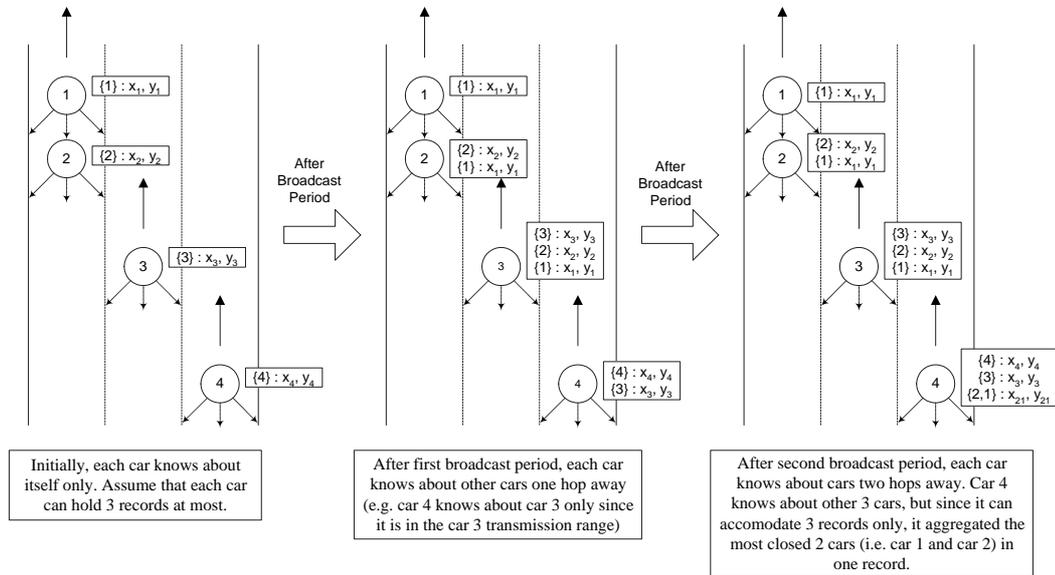


Figure 2: The problem this paper addresses (a) and the diffusion mechanism (b and c)

not scalable, due to messages flooding over the network, especially in high density roads.

In the other mechanism –the diffusion mechanism– each vehicle broadcasts information about itself and the other vehicles it knows about. Whenever a vehicle receives broadcast information, it updates its stored information and defers forwarding the information to the next broadcast period, at which time it broadcasts its updated information. The diffusion mechanism is scalable, since the number of broadcast messages is limited and no flooding is used. We use the diffusion mechanism in TrafficView.

As an illustration of the diffusion mechanism, assume for Figure 2(a), vehicles 2 and 3 are in the transmission range of vehicle 1. Likewise, vehicles 3 and 4 are in the range of vehicles 2 and 3, respectively. At the beginning, each vehicle knows only its own position and speed. After the first broadcast period (part (b) of the figure), vehicles 2 and 3 hear vehicle 1's broadcast about itself, and store such information. The same happens for vehicle 4 hearing vehicle 3's broadcast message. After the next broadcast period (part (c)), vehicle 4 hears the message broadcast by vehicle 3 which includes information about all of 1, 2, and 3, and updates its local information.

TrafficView does not suffer from memory limitation due to the small size of the stored records. As will be shown in Section IV, the average size for data records is on the order of 50 bytes. Assuming a very high density, five-lane road in which the distance between consecutive vehicles is 5 meters, about 5K bytes will be needed to store the information about all the vehicles in 100 meters, and about 1M bytes to store information of all the vehicles in 20Km. Most of the current portable devices come with more memory than these values.

On the other hand, assuming a transmission range of 250m for the wireless network card, there will be 50 vehicles competing for the same wireless medium in a single lane, and about 250 vehicles in a five-lane road assuming the lanes are close to each other. Hence, the total amount of data that needs to be broadcast by these vehicles every broadcast period is 250MB, which is beyond the capabilities of the current wireless technology. To cope with the bandwidth limitation, each vehicle is allowed to broadcast a small packet –a few kilobytes in size– every broadcast period to allow other surrounding vehicles to share the medium. Therefore, compression/aggregation mechanisms are needed to reduce the size of information to fit into the broadcast packet (node 4 in Figure 2(c)).

For simplicity, we assume throughout this paper that the road is straight. In the general case, the direction of the movement of a vehicle can be included in the record sent out about that vehicle, and then used to estimate its position on the road trajectory. Moreover, without loss of generality, we assume that the road is along the y axis, and all the vehicles are moving in the positive direction of the road. In a real situation, a road might be bidirectional, where vehicles move in two opposite directions. In this case, a vehicle will need to examine the movement vector in a record received about another vehicle, and ignore it if that vehicle is moving in the opposite direction. This can also be applied in the case of an intersection where a vehicle might hear about different vehicles moving in different directions.

IV. System Design

In this section we present the design of the implemented prototype of TrafficView system. Hereafter we use the



Figure 3: TrafficView prototype hardware components

terms “vehicle” and “node” interchangeably.

IV.A. Hardware

We implemented a prototype of the TrafficView system as shown in Figure 3. In this prototype, each vehicle is equipped with a portable computer (e.g., Compaq iPAQ with Linux Familiar distribution) augmented with two slots of PCMCIA sleeve, Global Positioning System (GPS), 802.11b wireless network card, DSP-100 2-port RS-232 serial PCMCIA card [1], and an OBDI-II interface [2]. The GPS receiver provides the latitude and longitude of the vehicle in addition to the global time. Using the wireless card, network connectivity is established, and the vehicle is able to send and receive information about other vehicles. The TrafficView software on the node periodically queries the vehicle’s status (e.g., speed) using the OBDI-II interface. The DSP-100 card is used to connect the iPAQ to the GPS receiver and the OBD-II interface.

IV.B. Software

In TrafficView, each vehicle stores records about itself and other vehicles it knows about. In this section, we describe the record format and the system modules.

IV.B.1. Data Representation

Each record about another vehicle consists of fields:

- *Identification (ID)*: Uniquely identify the records belonging to different vehicles.
- *Position (POS)*: The current estimated position of the vehicle.
- *Speed (SPD)*: Used to predict the vehicle’s position if no messages containing information about that vehicle are received.
- *Broadcast Time (BT)*: The global time at which the vehicle broadcast that information about itself.

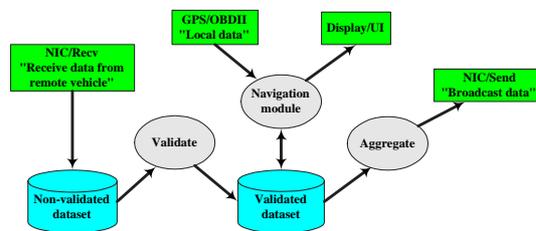


Figure 4: The structure of a node in TrafficView

IV.B.2. System Components

Figure 4 shows the software components (modules) of a node in the system. Each vehicle stores records about other vehicles in its local datasets. When the record is first received in 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 moved and merged with the *validated* dataset.

A TrafficView node, as shown in Figure 4, contains several modules that operate on its datasets:

- *GPS/OBD module* periodically updates the vehicle’s own record in the validated dataset. GPS readings are adjusted through the navigation module, which depends on GPS traces road maps formats, before storing them. For more information about navigation module, refer to [18].
- *Receive module* listens to broadcast messages from neighboring vehicles, and stores the records received in the non-validated dataset. It ignores the messages broadcast by its own vehicle.
- *Validation module* validates and resolves conflicts of the records in the non-validated dataset. It then merges the validated versions with the records in the validated dataset. For example, this module removes all the records that are about vehicles behind its own vehicle¹. Another example of a validity check is when there are multiple records containing information about the same vehicle. In this case, this module keeps the most recent record, and removes the older versions. In addition, this module periodically updates the estimated position of the vehicles in the validated dataset using the stored speeds. The validation module is also responsible of *information aging*, which will be discussed in Section V.D.
- *Aggregation module* performs aggregation algorithms on the records in the validated dataset in order to be able to place more information in the outgoing broadcast messages. This module might as well update the dataset by replacing the original records with the new aggregated version.

¹TrafficView only stores information about the vehicles in front of the current vehicle, and ignores the ones behind it.

- *Send module* writes the contents of the records in the validated dataset in a broadcast message and broadcasts it on the wireless channel using the wireless card.
- *Display/UI module* is responsible of displaying the validated records periodically on the display. It is also responsible for the user interaction (e.g., graphically and/or audibly).

V. Data Aggregation Mechanisms

A MAC layer protocol (e.g., IEEE 802.11b protocol) limits the size of the payload that is sent on the network channel to a maximum size (which is 2312 bytes for 802.11b). In TrafficView, the number of records in a node's validated dataset can be large, making it impossible to fit all of them in one broadcast message. In order to deliver as much information about other vehicles as possible, data compression/aggregation techniques should be applied to the validated records. Data compression and aggregation are two different concepts. Data compression is actually "binary compression" in the sense that it does not base the decisions made on the semantics of the data. Moreover, data compression techniques require a lot of computation resources which is not suitable for most portable devices. In this paper we focus on data aggregation mechanisms only.

Data aggregation is based on the data semantics. For example, the records from two vehicles can be replaced by a single record with little error if the vehicles are very close to each other, and they are moving with relatively the same speed. The way data aggregation contributes to the TrafficView system is by delivering as many records as possible in one broadcast message. This way, more new records can be delivered in certain period of time and the overall system performance is improved.

V.A. Data Aggregation Basics

A single aggregated record will represent information about a set of vehicles. In this paper we adopt one simple format for the aggregated records²: In an aggregated record, the ID field is extended to a list of vehicles' IDs while the other fields –position, speed, and broadcast time– remain as single values for all the vehicles stored in the record. Formally, if the records $(ID_1, POS_1, SPD_1, BT_1) \dots (ID_n, POS_n, SPD_n, BT_n)$ are being aggregated, and d_i is the estimated distance between the current vehicle and the vehicle with ID_i , the aggregated record will be

$$(\{ID_1, \dots, ID_n\}, POS_a, SPD_a, BT_a) \text{ where}$$

²We are developing other aggregation formats for the TrafficView system.

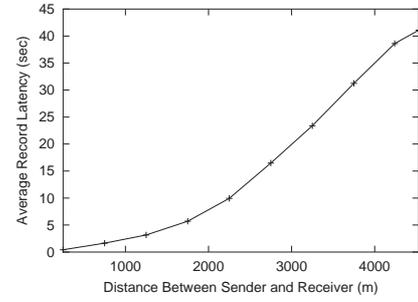


Figure 5: Average record delay based on the distance between the sender and receiver

$$\begin{aligned}
 POS_a &= \sum_{i=1}^n \alpha_i \times POS_i \\
 SPD_a &= \sum_{i=1}^n \alpha_i \times SPD_i \\
 BT_a &= \min\{BT_1, \dots, BT_n\} \\
 \alpha_i &= \frac{(\sum_{i=1}^n d_i) - d_i}{(n-1) \sum_{i=1}^n d_i}
 \end{aligned}$$

We realize that storing the minimum broadcast time –as opposed to storing the maximum or average– is advantageous, in that it allows the information about the vehicle which corresponds to the minimum broadcast time value to be updated as soon as a fresher record is heard about that vehicle.

According to the way the aggregated fields are calculated, the aggregated records should have close values to their POS , SPD , and BT fields to reduce the error resulting from the aggregation. Figure 5 shows the average difference between the record broadcast time and its receipt time, and the distance between the sender and the receiver, for a simulation of 550 total nodes, moving with an average speed of 30m/s, using the simple diffusion mechanism for information exchange with broadcast period of 2 seconds. As a result, if two records have close POS values, they are expected to have close BT values.

At the same time, if the difference between the speed of two vehicles that are close to each other is big, their distance will grow in a short time as well. Keeping in mind that the broadcast period is in the order of seconds, we can ignore the speed difference among the aggregated records, because the record will be updated with the new up-to-date position information as soon as new broadcast messages are heard. As a conclusion, the records are selected for aggregation based of their relative distances only. To achieve this in an efficient manner, records are kept sorted on the estimated relative distance of the current vehicle to the corresponding vehicles.

Whenever a node receives a record containing information about some vehicles, it first checks the information in that record against the validated records it has. If the record contains information about some vehicles which the node already knows, it performs the following:

1. If the broadcast time of the records is greater than the

ID	relative distance	speed	broadcast time
1	40	30	9.80
2	65	25	9.75
3	120	35	9.00
4	140	20	8.80
5	250	30	6.90
6	280	15	6.75
7	600	30	4.25

Table 1: Sample records used to illustrate different aggregation algorithms

broadcast time of the stored record, it means the new record is fresher, and therefore the node removes the corresponding vehicle ID from its stored record,

2. Otherwise, the new record contains older information, and hence the node removes the corresponding vehicle ID from the received record.

In TrafficView, vehicles apply the aggregation procedure on the records in the validated dataset each broadcast period to prepare the broadcast packet. Our preliminary experiments showed that the effect of each vehicle either replacing its current validated records with the aggregated version, or maintaining the original records in its validated dataset, on the quality of the information gained by other vehicles on the road, is almost identical; the only difference being the imposed overhead in the next broadcast period. We therefore decided to replace the validated dataset records with the new aggregated version during each broadcast period in order to reduce the overall aggregation overhead.

In the following subsections, we describe different algorithms to select records for aggregations. Table 1 lists a set of records that will be used for the illustration.

V.B. Ratio-based Algorithm

The algorithm divides the road in front of the vehicle to a number of regions (r_i). For each region, an aggregation ratio (a_i) is assigned. The aggregation ratio is defined as the inverse of the number of individual records that would be aggregated in a single record. Each region is assigned a portion (p_i where $0 < p_i \leq 1$) of the remaining free space in the broadcast message. The aggregation ratios and region portion values are assigned according to the importance of the regions and how accurate the broadcast information about the vehicles in that region is needed to be. For example, assigning decreasing values to the aggregation ratios and equal values to portion parameters will result in broadcasting less accurate information about regions that are farther away from the current vehicle, since for those regions, each individual record will represent large number of aggregated vehicles (records).

Algorithm 1: RATIO-BASED ALGORITHM()

INPUT :

Sorted list of validated records
 n : number of regions ($r_1 \dots r_n$)
 $a_1 \dots a_n$: aggregation ratios
 $p_1 \dots p_n$: message portion values

OUTPUT :

$th_1 \dots th_n$: merging thresholds
 $b_1 \dots b_n$: region boundaries

VARIABLES :

R : size of the remaining space in the broadcast message
 L : number of records left in the list of records
 $optimum$: optimum aggregation ratio
 d_{max} : distance of the farthest vehicle the current vehicle knows about
 l_i : number of records in region i

ALGORITHM :

main

Initialize b_i and th_i to 0 for all i

$b_0 \leftarrow d_{max}$

$R \leftarrow$ size of broadcast message

$L \leftarrow$ number of records in the input list

for each region r_i

$$\left\{ \begin{array}{l} optimum \leftarrow \frac{R}{(\text{average record size}) \times L} \\ \text{if } optimum \geq 1 \\ \quad \text{then return} \\ \text{if } optimum \geq a_i \\ \quad \text{then } \left\{ \begin{array}{l} b_i \leftarrow d_{max} \\ th_i \leftarrow \frac{b_i - b_{i-1}}{L \times optimum} \\ \text{return} \end{array} \right. \\ \text{do } \left\{ \begin{array}{l} l_i \leftarrow \text{number of records that fit in } \frac{R \times p_i}{a_i} \text{ bytes} \\ L \leftarrow L - l_i \\ \text{if } l_i = 0 \\ \quad \text{then } \left\{ \begin{array}{l} b_{i-1} \leftarrow d_{max} \\ \text{return} \end{array} \right. \\ b_i \leftarrow \text{relative distance of the last record fit} \\ th_i \leftarrow \frac{b_i - b_{i-1}}{l_i \times a_i} \\ R \leftarrow R - R \times p_i \end{array} \right. \end{array} \right.$$

Given the aggregation ratios, portion values, and number of regions, the algorithm calculate the region boundaries ($[b_i, b_{i+1}[$) as shown in Algorithm 1. Knowing the number of current records in the validated dataset that lie within the boundaries of each region and the corresponding free space in the broadcast packet, the algorithm calculates the merging threshold (th_i) corresponding to each region. Any set of consecutive records in region r_i will be aggregated in a single record if the relative distance (in y direction) between the first and the last record is less than the corresponding merge threshold, th_i .

As shown in Algorithm 1, the algorithm will not over-aggregate the records. This is guaranteed by calculating the optimum aggregation ratio at the beginning of the loop for each region. This aggregation ratio is the value

ID(s)	relative distance	speed	broadcast time
1, 2, 3	67.56	29.39	9.00
4, 5, 6	215.22	21.68	6.75
7	600	30	4.25

Table 2: Records sent out by the Ratio-based algorithm

needed to fit the rest of the records in the message free space. If this ratio is greater than or equal to one, the algorithm terminates since no aggregation is needed. Otherwise, the optimum value and the aggregation ratio of the current region are compared and the maximum among these two is used.

After the algorithm aggregates the records, it starts writing the record contents to the broadcast message until no free space is left. There is no guarantee to write all the record contents in the message. The tradeoff between the number of records written and the accuracy of the records is governed by the used parameter values.

As an example, assume a vehicle with $ID = 0$, using this algorithm, divides the road into two regions, and the corresponding parameter are $a_1 = 0.5$ with $p_1 = 0.5$ and $a_2 = 0.25$ with $p_2 = 0.5$. If the algorithm is applied to the records of Table 1, it will calculate the parameters: $b_1 = 120$, $th_1 = 80$, $b_2 = 600$, and $th_2 = 261.8$. Note that th_2 is calculated using the optimal aggregation ratio 0.46 instead of the input value, 0.25.

After calculating the parameters, in the first region, the algorithm first combines records 1 and 2, and then combines the result with record 3. Likewise, the records 4, 5, and 6 are combined in the second region. The records sent out by the algorithm are shown in Table 2. Record 7 is sent not aggregated.

V.C. Cost-based Aggregation

In the Ratio-based algorithm, records that satisfy the merging threshold, (th_i), criterion are “blindly” combined without considering the cost of the aggregation. In contrast, the Cost-based algorithm assigns a cost for aggregating each pair of records, and whenever it needs to aggregate two records, the two that correspond to the minimum cost are chosen. Assume two records storing aggregated information about s_1 and s_2 number of vehicles, with a relative distance of d_1 and d_2 , respectively. The cost of aggregating the two records is calculated as follow:

$$cost = \frac{|d_1 - d_a| \times s_1 + |d_2 - d_a| \times s_2}{d_a}$$

where d_a is the relative distance of the aggregated group of records (vehicles). This formula is calculated such that it: 1) assigns a high cost for the vehicles that are relatively close to the current vehicle ($1/d_a$), 2) tries to minimize the error introduced during the merging ($|d_i - d_a|$), and

Algorithm

2: COST-BASED AGGREGATION()

INPUT :

Sorted list of validated records
 $cost_threshold$
 n : number of regions ($r_1 \dots r_n$)
 $a_1 \dots a_n$: aggregation ratios
 $p_1 \dots p_n$: message portion values

VARIABLES :

R : size of the remaining space in the broadcast message
 L : number of records left in the list of records
 $optimum$: optimum aggregation ratio
 l_i : number of records in region i

ALGORITHM :

main

$R \leftarrow$ size of broadcast message

$L \leftarrow$ number of records in the input list

for each region r_i

```

do {
  optimum ←  $\frac{R}{(\text{average record size}) \times L}$ 
  if optimum ≥ 1
    then return
   $a_i \leftarrow \max(\text{optimum}, a_i)$ 
  goal ←  $a_i \times L$ 
  while  $L > \text{goal}$ 
    do {
       $c \leftarrow$  minimum cost of merging two consecutive
        records in the remaining records set
      if  $c > cost\_threshold$ 
        then return
      Merge the two records corresponding to the
        minimum cost
       $L \leftarrow L - 1$ 
    }
   $l_i \leftarrow$  number of records that fit in  $R \times p_i$  bytes
   $R \leftarrow R - \text{size of the } l_i \text{ records}$ 
}

```

3) minimizes the number of vehicles affected by the aggregation (s_i).

The details of the algorithm are shown in Algorithm 2. The aggregation ratios and message portion values are the inputs to the algorithm. For each aggregation ratio and the corresponding portion value, the algorithm starts by continuously selecting the two records that result in the minimum cost, and aggregating them until the number of records is reduced to the value needed by the factor of the aggregation ratio. Afterwards, it writes the contents of the first records in the sorted list to the beginning of the message until they fill the space allocated according to the corresponding portion value. In the next iteration, the same procedure of aggregation and writing is applied to the rest of the records that are not written yet. The aggregation ratios in each iteration is compared with the optimum aggregation ratio to avoid over-aggregation.

A problem that might happen is that as the algorithm proceeds, the number of records left decreases, and the distance between any two consecutive records increases. Hence there is a risk of combining two records that

ID(s)	relative distance	speed	broadcast time
1, 2	49.52	28.09	9.75
3, 4	129.23	28.07	8.80
5, 6	264.15	22.92	6.75

Table 3: Records sent out by the Cost-based algorithm

correspond to vehicles that are too far away from each other. To avoid this problem, the algorithm terminates as soon as the calculated cost is greater than a threshold parameter (*cost-threshold*.)

For example, assume vehicle with $ID = 0$ intends to use this algorithm for the records listed in Table 1, where $a_1 = a_2 = 0.5$, $p_1 = p_2 = 0.5$, and *cost-threshold* = 0.9. During the first iteration (a_1), it first aggregates records 5 and 6 (cost = 0.11), then 3 and 4 (cost = 0.15), and finally 1 and 2 (cost = 0.50). In the second phase (a_2), the minimum cost is 1.22, which is greater than the cost threshold, therefore the algorithm terminates. Table 3 lists the records that are sent out by vehicle 0 and the corresponding fields. In this case, vehicle 0 cannot fit record 7 in its message.

V.D. Information Aging

The records stored in both the validated and non-validated datasets, must be examined to verify that they reflect the current state of the road and eliminate any outdated (old) information. For example, vehicles included in the validated dataset might have exited the road. Moreover, new received records (non-validated) might contain inaccurate information due to frequent changes in the speed of the corresponding vehicles and/or aggregation mechanisms applied to the data within relaying nodes.

There are two main problems here: how should the value of the information in a broadcast message be assessed, and how can a balance between knowing inaccurate information about a vehicle, and having no knowledge about it, be achieved. In general, if the cost of knowing inaccurate information about vehicle j that is at a relative distance of d is a function $c_1(j, d)$, and the cost of having no information about j is another function $c_2(j, d)$, the information should be accepted and stored if $c_1(j, d) < c_2(j, d)$, otherwise it should be dropped. Unfortunately, it is not clear how to assign values to these two functions.

To solve this problem, TrafficView exploits two aging mechanisms. The first mechanism associates a timer with each record added to the validated dataset. This timer is reset each time the record is updated by a broadcast message. If the timer is expired, the record is dropped. The second mechanism, which we call Receive-aging, deals with newly received records via broadcast messages. Whenever a new record is received, the

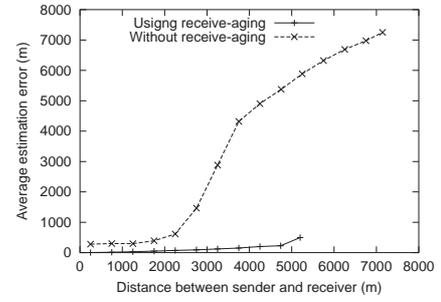


Figure 6: Effect of Receive-aging: average error with/without Receive-aging mechanism

expected latency in receiving the record is calculated and compared to the actual latency (the difference between the receive time and the BT field.) If the difference between these two is lower than a threshold, it is stored; otherwise, it is considered out-of-date, and is ignored.

Formally, assume node 2 receives a record about vehicle 1 at time t . Looking at the record contents, node 2 extracts the time BT_1 at which the record was first broadcast, and vehicle 1's position POS_1 at that time. Knowing its own position POS , node 2 estimates its position POS_2 at time BT_1 as

$$POS_2 = POS - v_2 \times (t - BT_1)$$

where v_2 denotes node 2's speed which we assume, with no loss of generality, to be fixed during the time period $[BT_1, t]$. Node 2 then calculates the expected delay in receiving the record as:

$$delay = \frac{|POS_1 - POS_2|}{|r/p + v_2|}$$

where r is the wireless transmission range, and p is the broadcast period. Therefore, r/p is the approximate propagation speed of the information between the vehicles. This record is then accepted by node 2 only if

$$|t - BT_1| \leq \delta_1 + (1 + \delta_2) \times delay$$

where δ_1 and δ_2 are acceptance thresholds.

To validate the effectiveness of the Receive-aging mechanism, we ran two simulations with 870 total nodes moving with an average speed of 30m/s. In the first run, the nodes were using this mechanism with $\delta_1 = 6.0$ and $\delta_2 = 0.3$, whereas in the second, it was disabled. Figure 6 presents average estimation error of the position of the vehicles in the two runs for different distance between the sender and receiver. As shown, when Receive-aging is not used, the estimation error for vehicles at far away distances is huge. In contrast, using this mechanism has reduced the average error to a small value.

VI. Performance Evaluation

We have implemented our mechanisms in ns-2 simulator to compare the performance of different algorithms.

In this section, we present the experiments, and the corresponding results. In addition, we evaluated the prototype using real GPS traces obtained on a highway.

VI.A. Scenario Generator

Modelling road traffic is a research topic about which a lot of work has been done. For example CORSIM [6] is a microscopic traffic simulator developed by The Federal Highway Administration. Unfortunately, none of the traffic modeler tools are freely available to public. We have therefore developed our own scenario generator tool based on “setdest”—a generator tool for random-way point mobility model, developed at Carnegie Mellon.

The scenario generator accepts as parameters simulation time, road length, nodes average speed, number of lanes on the road, and the average gap length between vehicles. It uses a simplified traffic model as follows:

- *Entries and Exits*: The entries and exits are evenly distributed along the road each 1000 meters. Vehicles may enter the road at each entry except the last one and leave at any subsequent exit. Vehicles enter the road at the front-end entry with a probability of 0.7, and at side entries with a probability of 0.3.
- *Speed Changes*: To model the changes to the node’s speed, the road between the entry point and exit point of a node is divided into regions of 50 meters, and a constant speed of $\text{max speed} \times (0.75 + \text{rand}(-2, 2) \times 0.125)$ is used for each region, where $\text{rand}(a, b)$ returns a uniformly distributed random integer between a and b .
- *Changing Lanes*: Vehicles can change their lanes with no dependence on other vehicles. The probability of staying on the same lane is 0.6 whereas the probability of changing to the right or left lane is 0.2.
- *Vehicle Density*: The density of vehicles is an important factor because it determines the number of neighboring nodes in the transmission range of a vehicle, which has a great impact on the transmission delay and available bandwidth of the network. The scenario generator initially puts

$$\frac{\text{road-length} \times \text{number of lanes}}{\text{average gap}}$$

active nodes, evenly distributed, on the road. Once a vehicle leaves the road at one of the exits, it is deactivated, and a new node is added (activated) to the road randomly. As soon as a node is deactivated, it will no longer affect our metric calculations introduced in the next section.

Figure 7 shows the histogram of the average speed and number of lane changes per minute for a scenario generate with average speed = 30m, and average gap =

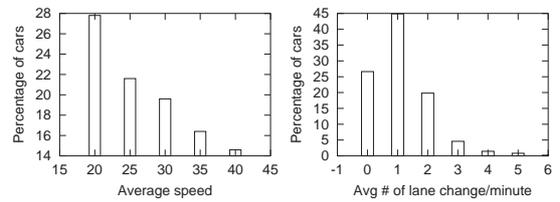


Figure 7: Sample histograms of average speed (left) and average number of lane changes per minute (right) in a scenario generated by the scenario generator tool

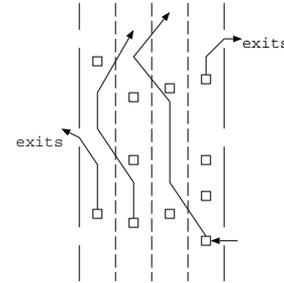


Figure 8: A segment of a road in an example scenario generated by the scenario generator

100. The graphs show the percentage of vehicles that have that average speed and average number of lane changes per minute, respectively. A segment of a road in an example scenario generated by the tool is shown in Figure 8. The road, along which 11 nodes are moving, has three exits at each side.

For all the simulations in this paper, we fixed the length of the road to be 15,000 meters with 4 lanes. We used 802.11b (with a data transmission rate of 11Mb) as the wireless media with a transmission range of 250m³. During a simulation, nodes broadcast messages periodically. The broadcast period is selected uniformly from [1.75, 2.25] seconds, and each node recalculates the next broadcast period after the current broadcast. For all the simulation runs, we use broadcast messages of size 2312 (the maximum payload size of 802.11b standards) and we fix the simulation time to 300 seconds.

VI.B. Algorithms and Metrics

We implemented two simple algorithms in addition to the ones introduced in Section V for comparison purposes: non-aggregation and brute-force cost-based. In the non-aggregation method, no aggregation is performed and each node broadcasts only the first records in its validated dataset that fit in one broadcast message. In the brute-force cost-based algorithm, the node keeps aggregating its records using the same technique introduced in the Cost-based algorithm, until it can fit all the its records in one broadcast message.

³In practice, we found out that the wireless transmission range is less than 250m. However, using external antennas, we can restore this transmission range.

We will use the following metrics and graphs to assess the performance of the algorithms:

- **Accuracy:** The road in front of each vehicle is divided into regions of 500 meters long, and the average error in estimating the position of vehicles in each region is calculated. In the accuracy graphs, the average estimation error for each region is shown, averaged over all the nodes during the simulation.
- **Visibility:** We define the visibility of a specific vehicle as the *average* relative distance to the vehicles it knows about. A point (d, p) on a visibility graph means that $p\%$ of the vehicles have had a visibility of d meters or more.
- **Knowledge Percentage:** The road in front of each vehicle is divided into regions of 200 meters long. For each region, the percentage of the vehicles in that region about which the current node knows, is defined as the knowledge percentage of that node for that region. The knowledge percentage graph presents the knowledge percentage for each region, averaged over all the nodes during a simulation run.

VI.C. Aggregation Parameters

We ran different simulations to select the suitable values for the parameters of the Ratio-based and Cost-based algorithms with total number of 960 nodes and average speed of 30m/s. The suitable set of values are used in the runs to compare the performance of different algorithms.

For the aggregation algorithms, the maximum number of regions in front of each node is four. The first three regions are defined by parameters a_1 , a_2 , a_3 , p_1 , p_2 and p_3 . The fourth region is defined dynamically by the remaining available space in the outgoing message and the remaining set of records that each node has.

Table 4 lists the parameters used in different runs of the algorithms. The way these parameters are selected is to first run the algorithm with param1, and param2 to select the better a_i values and then fix a_i and run with param3 and param4 to choose p_i values. The incentive is to select a_i as small as possible to achieve as large visibility as possible while maintaining a good accuracy for the closer vehicles. The reason we started with the a_i values is that they have a larger effect on the performance of the aggregation algorithms than the effect of p_i parameters. Figure 9 shows the visibility graph for different runs of the Ratio-based algorithm. We found out that param1 settings give a higher accuracy while maintaining a good visibility. We therefore use param1 values to set the Ratio-based parameters in the rest of the simulation runs. On the other hand, we noticed that using param4 gives a higher accuracy among the other settings for the Cost-based aggregation algorithm while maintaining a good

Name	a_1	a_2	a_3	p_1	p_2	p_3
param1	0.5	0.25	0.17	0.5	0.5	0.5
param2	0.75	0.5	0.25	0.5	0.5	0.5
param3	0.5	0.25	0.17	0.4	0.6	0.8
param4	0.5	0.25	0.17	0.3	0.43	0.75

Table 4: Parameter settings for different runs of the Ratio-based and Cost-based aggregation algorithms

Name	Total nodes	Avg. speed	Avg. gap
Rush-hour	690	10	100
City	780	20	100
High-density highway	870	30	100
Low-density highway	548	40	175

Table 5: Parameters of different simulations used to compare different algorithms

visibility as shown in Figure 10. We therefore use the values of param4 in the rest of the simulation runs of the Cost-based algorithm.

For the Receive-aging mechanism, we set δ_1 to 6.0 and δ_2 to 0.3. These values were selected by running the non-aggregation method with different values for these parameters, and choosing the ones that resulted in the best visibility while maintaining an acceptable accuracy.

VI.D. Results

To compare the performance of different algorithms, we ran each algorithm for different scenarios. Table 5 lists the configuration of each simulation scenario.

We first look at the effect of the road parameters. Figure 11 shows the visibility graph for runs on different scenarios of the non-aggregation algorithm. We notice in this Figure that average speed does not have a significant effect on the performance of the algorithm. On the other hand, the average gap, directly effects the performance: As the gap between vehicles increases, the number of vehicles scattered over the road decreases. Therefore, the broadcast message will contain records about vehicles in farther distances and thus it increases the visibility.

Figure 12 shows the same graph for the brute-force algorithm. For this algorithm, as the average speed increases, the rate of vehicles get closer to or depart from each other increases. Therefore, more number of records get aggregated. With the increase in cars speed, the values of broadcast fields (*BT*) fields decrease faster and that result in invalidating records more quickly due to aging mechanisms, and hence the average visibility decreases. Again, increasing the gap value increases the vehicles visibility. The other aggregation mechanisms show a similar behavior. We use *High-density highway* scenario for performance comparison between different aggregation algorithms.

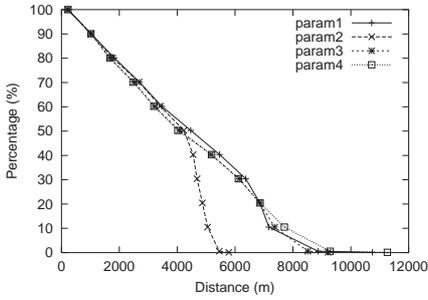


Figure 9: Visibility graphs for Ratio-based using different aggr. parameters

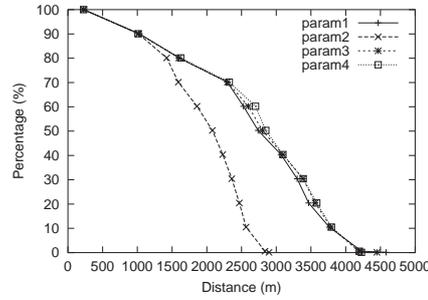


Figure 10: Visibility graphs for Cost-based using different aggr. parameters

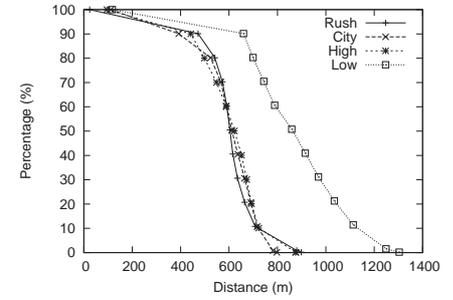


Figure 11: Visibility graphs for Non-aggregation using different scenarios

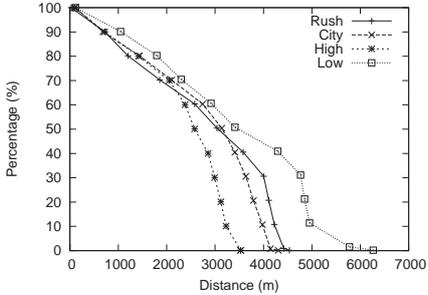


Figure 12: Visibility graphs for Brute-force using different scenarios

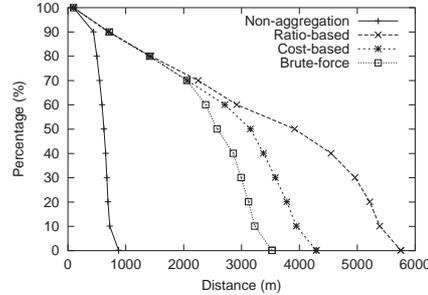


Figure 13: Visibility graphs for different aggregations using *High scenario*

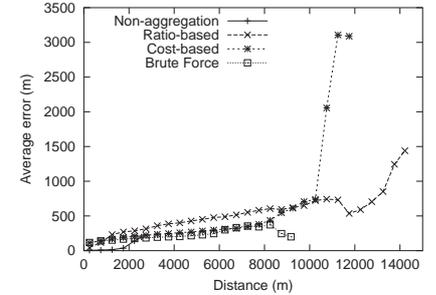


Figure 14: Average error for different aggregations using *High scenario*

Figure 13 shows the visibility graph of the different algorithms. The Ratio-based algorithm achieved the highest visibility value. The Cost-based algorithm outperforms the brute-force algorithm. As mentioned earlier, this is due to the fact that records are invalidated more quickly in the brute-force algorithm. The reason the Ratio-based achieves the highest visibility is that it performs aggregation on all the vehicles in all the regions while the Cost-based and brute-force methods have less or no control on selecting the region where the aggregation is performed. The result indicates that the boundaries of the regions generated by Ratio-based algorithm cover larger road areas than the other algorithms, and hence it has the highest visibility.

Figures 14 and 15 present average estimation error and average knowledge percentage for different algorithms using *High-density highway* scenario. As a result of the Ratio-based mechanism performing aggregation on all the regions, its knowledge percentage about the close and medium-distanced vehicles is less than the other algorithms; its accuracy is also lower than the other algorithms.

Next, we present the evaluation of the performance of our prototype using real GPS traces obtained on a highway. In doing this, we have acquired eight 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. We fed these traces, as movement patterns for eight vehicles, to the TrafficView prototype. We measured the performance of the prototype in terms of visibility and accuracy achieved by the *ratio-based* aggregation versus non-aggregation algorithms.

Although our experiments used a small number of vehicles, the effect of the ratio-based aggregation is still significant compared to the non-aggregation case. Figure 16 shows the maximum vehicle visibility along the road. For non-aggregation case, all cars have maximum visibility of at least 300m ahead, whereas about 25% of the cars have visibility of at least 525m. This percentage increases for the aggregation case, where about 75% of the cars have a visibility for more than 525m.

From Figure 17, 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 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.

From the above results we conclude that the Ratio-based algorithm is more flexible than the other algorithms in that it provides more control over the tradeoff between the accuracy and visibility governed by the parameter setting. For the other methods, although tuning the parameters is easier, the cost function does not provide the flexibility present in the Ratio-based algorithm.

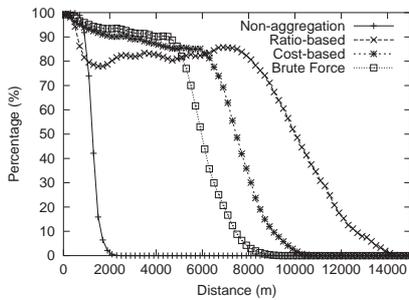


Figure 15: Average knowledge for different aggregations using *High scenario*

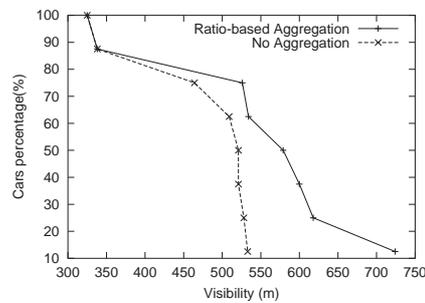


Figure 16: Visibility graphs using eight real GPS traces

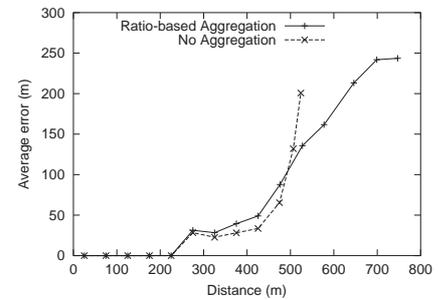


Figure 17: Average position error using eight real GPS traces

VII. Conclusions and Future Work

In this paper we introduced the TrafficView system, which is a part of broader project—e-Road—that is still under development. The goal of TrafficView is to provide the driver of a vehicle with information about traffic and road conditions. The essence of the system is to gather and disseminate traffic information between the vehicles on the road. We presented the basic design of the system, and the algorithms used for data aggregation and information dissemination using the 802.11b standards.

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 others to gain information about the vehicle without identifying it.

Security and trust are two other important issues in such a system. A fraudulent vehicle could disseminate information about nonexistent vehicles, or broadcast bogus information about existing vehicles. Different mechanisms should be proposed to prevent this and to identify those fraudulent vehicles to avoid them.

For future work, we are continuing to work in a number of different directions as the privacy and the security issues. We are experimenting with a linear programming model to estimate the aggregation parameters dynamically based on the road condition. We believe that TrafficView and the e-Road project will greatly enhance and ease the driving experience. At the same time, they will encourage and trigger several applications to be built over these systems.

References

- [1] <http://www.quatech.com>
- [2] <http://www.obddiagnostics.com>
- [3] L. Briesemeister, G. Hommel, "Role-based multicast in highly mobile but sparsely connected ad hoc networks," in *First Annual Workshop on Mobile Ad Hoc Networking and Computing*, pp. 45–50, Aug. 2000.
- [4] Z. D. Chen, H. T. Kung, D. Vlah, "Ad Hoc Relay Wireless Networks over Moving Vehicles on Highways," in *Proceeding of the ACM International Symposium on Mobile Ad-hoc Networking and Computing*, Oct. 2001.
- [5] I. Chisalita, N. Shahmehri, "A peer-to-peer approach to vehicular communication for the support of traffic safety applications," *5th IEEE Conference on Intelligent Transportation Systems*, Singapore, Sept. 2002.
- [6] CORSIM User Manual, Ver. 1.01, The Federal Highway Administration, US Dept. of Transportation, 1996.
- [7] General Motors Collaborative Laboratory website available online at <http://gm.web.cmu.edu/>.
- [8] W. Kellerer, "(Auto)Mobile Communication in a Heterogeneous and Converged World," *IEEE Personal Communications*, Vol. 8(6), pp. 41–47, Dec. 2001.
- [9] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing," *ACM Mobicom 2000*, Boston, MA.
- [10] Q. Li, D. Rus, "Sending Messages to Mobile Users in Disconnected Ad-hoc Wireless Networks," *ACM Mobicom 2000*, Boston, MA.
- [11] R. Miller, Q. Huang, "An Adaptive Peer-to-Peer Collision Warning System," *IEEE Vehicular Technology Conference (VTC)*, Birmingham, AL, May 2002.
- [12] R. Morris, J. Jannotti, F. Kaashoek, J. Li, D. Decouto, "CarNet: A Scalable Ad Hoc Wireless Network System," *9th ACM SIGOPS European Workshop*, Kolding, Denmark, Sept. 2000.
- [13] P. Murphy, E. Welsh, P. Frantz, "Using Bluetooth for Short-Term Ad-Hoc Connections Between Moving Vehicles: A Feasibility Study," *IEEE Vehicular Technology Conference (VTC)*, Birmingham, AL, May 2002.
- [14] Q. Huang, R. Miller, "The Design of Reliable Protocols for Wireless Traffic Signal Systems". Technical Report WUCS-02-45, Washington University, Department of Computer Science and Engineering, St. Louis, MO.
- [15] M. Satyanarayanan, "Pervasive Computing: Vision and Challenges," *IEEE Personal Communications*, Vol. 8(4), pp. 10–17, Aug. 2001.
- [16] C. Schwingenschloegl, T. Kosch, "Geocast Enhancements for AODV in Vehicular Networks," *ACM Mobile Computing and Communications Review*, Vol. 6(3), pp. 96–97, Jul. 2002.
- [17] E. Welsh, P. Murphy, P. Frantz, "A Mobile Testbed for GPS-Based ITS/IVC and Ad Hoc Routing Experimentation," *International Symposium on Wireless Personal Multimedia Communications (WPMC)*, Honolulu, HI, Oct. 2002.
- [18] S. Dashtinezhad, T. Nadeem, B. Dorohonceanu, C. Borcea, P. Kang, L. Iftode, "TrafficView: A Driver Assistant Device for Traffic Monitoring based on Car-to-Car Communication," *Proceedings of the IEEE Semiannual Vehicular Technology Conference*, Milan, Italy, May 2004.