A cooperative scheme to aggregate spatio-temporal events in VANETs

Dorsaf Zekri
Institut TELECOM
TELECOM SudParis
UMR CNRS SAMOVAR
Evry, France
Dorsaf.Zekri@it-sudparis.eu

Bruno Defude
Institut TELECOM
TELECOM SudParis
UMR CNRS SAMOVAR
Evry, France
Bruno.Defude@it-sudparis.eu

Thierry Delot
University Lille North of France
CNRS LAMIIH
INRIA Lille Nord Europe
Valenciennes, France
Thierry.Delot@inria.fr

ABSTRACT

Today, thanks to vehicular networks, drivers may receive useful information produced or relayed by neighboring sensors or vehicles (e.g., the location of an available parking space, of a traffic congestion, etc.). In this paper, we address the problem of providing assistance to the driver when no recent information has been received on his/her vehicle. Therefore, we present a cooperative scheme to aggregate, store and exchange these events in order to have an history of past events. This scheme is based on a dedicated spatio-temporal aggregation structure using Flajolet-Martin sketches and deployed on each vehicle. Contrary to existing approaches considering data aggregation in vehicular networks, our main goal here is not to save network bandwidth but rather to extract useful knowledge from previous observations. In this paper, we present our aggregation data structure, the associated exchange protocol and a set of experiments showing the effectiveness of our proposal.

Keywords
VANETs, event streams, sensor data, spatio-temporal event, event aggregation.

1. INTRODUCTION

Nowadays, there is a great interest in developing systems to assist drivers on the road. Therefore, vehicular ad hoc networks (VANETs) have recently received a lot of interest since they provide a way to improve both safety and sustainability. VANETs rely on the use of short-range networks (about a hundred meters), for vehicles to communicate [10]. Using such communication networks, the driver of a car can receive information from its neighbors or from the roadside infrastructure (if any). Many pieces of information can be exchanged in the context of inter-vehicle communications, for instance to warn drivers when a potentially dangerous event arises (e.g., an accident, an emergency braking, an obstacle on the road, etc.) or to try to assist them by providing relevant information (e.g., about traffic congestions, real-time traffic conditions, close available parking spaces, etc.). Our work takes place within the VESPA project [3], a system designed for vehicles to share information in inter-vehicle ad-hoc networks. The originality of VESPA is to support the exchange of any type of event in the network (e.g., traffic congestion, emergency braking, available parking space, etc.) thanks to an adaptive dissemination protocol [1].

Even if they may provide an effective assistance to the driver in many cases, those systems exploiting vehicular networks to gather relevant information for the driver also suffer from a major drawback.

Indeed, with VESPA and similar systems based on events dissemination, messages representing physical events (e.g., a traffic congestion, a parking space released, etc.) are generated and exchanged between vehicles. Then, once these messages have been possibly used to warn/inform the driver, they are considered obsolete and deleted. On the contrary, our objective in this work is to store and summarize the information associated to the events previously observed. More precisely, we keep for each (type of) event the spatial area where it occurred and a time window characterizing the generation of the event. Then, we exploit the summarized information to extract useful knowledge for the driver and provide assistance to him/her when needed, even if no information has been transmitted by other neighboring nodes. For example, knowing the frequency of parking spaces released in different spatio-temporal area, it becomes possible to determine the one(s) with the highest probability of finding free places at a given day and hour. In another context, when many accidents or emergency braking are observed in a particular geographical area, it is possible to conclude that this area is dangerous enough to warn drivers.

The rest of this paper is organized as follows. Section 2 describes our cooperative scheme. Section 3 focuses on the
spatio-temporal aggregation data structure while section 4 defines our exchange protocol. Section 5 presents the experimental evaluation of our proposal. In section 6, we compare our approach with related works. Finally, we conclude and present the perspectives of our work in section 7.

2. A COOPERATIVE SCHEME TO AGGREGATE EVENTS

2.1 Principles

In this paper, we consider smart vehicles able to provide both alert services and decision support to drivers. Thus, a decision support system for drivers of a vehicle $i$ can acquire information about events observed either by itself (e.g., using embedded sensors) or diffused by other vehicles. Obviously, the information available on each vehicle is partial since it cannot perceive all events or receive all messages transmitted by the other vehicles using short-range wireless networks. Vehicles can also acquire information from a fixed infrastructure deployed along the roads. In urban areas, the infrastructure may be a central parking management system providing information to vehicles driving in its vicinity.

In order to avoid losing information related to the events observed, we propose to add a new source of knowledge based on the aggregation of obsolete events (i.e., possibly used to produce a warning to the driver). Our objective is to keep a summary of previous messages to estimate whether an event can happen, even without any further observation.

Obviously, there exist a variety of techniques that can be used to build summaries of spatio-temporal events. In our case, the important criteria expected for a summary are (i) to estimate the frequency of (type of) event occurrences; (ii) to promote basic dimensions such as location and time; (iii) to be incrementally constructible and inexpensive in both computing time and storage space; (iv) to let each driver define the types of events s/he is interested in, as well as the spatial and temporal scales s/he wants to use for the aggregation process; (v) to allow the exchange of (parts of) summaries between vehicles in order to enrich their knowledge base. Indeed, each vehicle is only notified of a small subset of occurred events. It can so benefit from the observations that other vehicles have summarized.

3. SPATIO-TEMPORAL AGGREGATION DATA STRUCTURE

We assume that each vehicle $V_i$ observes a set of events $E$. Each event $e$ of $E$ is characterized by:

1. $t_e$: the type of event (e.g., accident, parking space, etc.).
2. $loc_e$: the location of the event and its timestamp. This information is provided by GPS like positioning systems. Note that the use of the GPS system also allows each vehicle to share the same clock and avoid synchronization issues.

3. $id_e$: the unique event identifier $e$. This unique identifier is necessary for the detection of duplicates. Here, we assume that an instance of event is allocated a unique identifier. When the event occurs, such a unique identifier can be generated by combining the current time and the GPS location of the event with a randomly-generated sequence.\footnote{The generation of a unique identifier for events observed by several vehicles (e.g., different vehicles stuck in a traffic congestion) is still an open issue. Interesting ideas to solve it have been proposed in the field of information fusion [4, 7].}

To allow the summarization of events and facilitate the exchange of summaries without loss of information, we organize the spatio-temporal area in two distinct levels. The physical level represents a fixed partition of the area in squares for the spatial dimension and in intervals for the temporal dimension. All the vehicles share this partition of space and time and so the physical level is the same for all of them. To allow drivers to decide which spatial areas are of interest for them, we add a logical level on top of the physical one. A logical cell (or interest area) is defined as a set of physical cells.

In the rest of this paper, we assume the following notations. The physical space is divided into $CnP$ squares ($N$ squares on the X axis and $P$ ones on the Y axis) with $g$ temporal granularities for each day so $7 \times g$ for a week. The coordinates of the origin point are $(x_{\text{origin}}, y_{\text{origin}})$. An interest area is defined by a pair of physical cells. Coordinates $(i, j)$ of the bottom left cell and the coordinates $(k, l)$ of the upper right one define this interest area. We share the same temporal granularities at the logical level and at the physical level. For example, we can use 84 temporal granularities ordered from Monday from 0:00AM to 2:00AM to Sunday 10:00PM to 12:00PM. The data structure representing this spatio-temporal organization is illustrated in Figure 1. It allows a quick access to the interest areas according to spatial coordinates and consists of an ordered array of interest areas (identified by an id) and bounded by two physical cells.

In the example presented in Figure 1, the interest area with id 4 is delimited by both physical cells with coordinates (90,130) and (120,150). Available parking spaces are observed for this area. The array of interest areas is sorted by increasing values of i. Each interest area is described by a linked list representing the physical cells it contains (6 cells in our example). For each cell, we finally store $7 \times g$ elements (according to the defined temporal granularity) containing the frequency of observed events for the corresponding physical cell for a given time granularity. We estimate the frequency of events as the ratio between the number of observed events (maintained by an event sketch) and the number of observed weeks (maintained by a timestamp sketch). For example, if 20 events are observed during 4 weeks, the frequency is set to 5.
To summarize the information about the events known by each vehicle, we use Flajolet-Martin sketches [5]. Flajolet-Martin sketches provide a compact representation to estimate the number of occurrences of distinct events. In addition to the limited storage space they require, sketches are particularly interesting in our context because they allow detecting duplicates by construction. Indeed, as we will explain in detail in section 4, the information aggregated on each vehicle can be exchanged and merged with other ones. Indeed, the quality of the predictions realized thanks to the summaries (e.g., where is the area with the highest probability to find an available parking space?) depends on the amount of data contained in these summaries.

Therefore, for obvious storage space reasons, each vehicle can not keep the identifier of all observed events. Besides, by using simple counters instead of the sketches in the aggregation data structure to estimate the frequency of events (i.e., counting the event of each type occurred in a particular spatio-temporal zone), as proposed in [2], it is not possible any more to determine whether the observations done by several vehicles correspond to the same physical event or not.

On the contrary, since different instances of the same event observed by different vehicles have the same image computed by the hash function, this problem is avoided when using Flajolet-Martin sketches.

A sketch is a set of r bits initially set to 0. The size of the sketch is defined according to the size of the considered set of items. Sketches are constructed by applying hash functions to items (e.g., to the identifier of the event for our event sketch). Thus, for each object o to store, the \( h(o) \)th bit is set to 1 where \( h \) is the hash function applied.

Flajolet-Martin sketches are associated with probabilistic counting algorithms. Thus, when items have been inserted into a sketch, the number of distinct objects contained can be estimated by \( n = 1,29 \times 2^k \) (with \( k \) the position of the first bit in the sketch that is still set to 0) [5]. To increase the accuracy of the estimation, \( m \) hash functions can be applied to produce \( m \) distinct sketches (and not just one). To minimize the cost, one can randomly choose one hash function to be applied on each item. In this case, the number of items will be evaluated by the sum returned by each sketch. The standard error is \( O(m^{-1/2}) \), so with \( m = 4 \), we get a good precision.

In our aggregation data structure, the event sketch (estimating the number of observed events) is constructed by applying a hash function on the identifier \( id_e \) of each event observed. As concerns the timestamp sketch (estimating the number of weeks observations), it is constructed by applying a hash function on the number of the week. The size of a sketch depends on the maximum number of items it should contain. In our case, we consider that we have to store up to 1,000,000 events of a particular type in a cell and so 20 bits are required for each event sketch. Moreover, about 256 weeks and so 8 bits are needed per timestamp sketch. To resume, \( m=4 \) sketches of size \( k=20 \) bits for the events and \( m=4 \) sketches of size \( k=8 \) bits for weeks are stored for each temporal granularity of each physical cell. Let us assume that a vehicle observed \( P \) interest areas composed each one by \( M \) physical cells with \( E \) types of events aggregated over all temporal granularities. All these observations are done on all week (7 days). The aggregation structure size is:

\[
Size = P \times \left( (id + i + j + k + l) \text{ bytes} + E \text{ bits} + E \text{ pointers} \right) + E \times M (i + j) \text{ bytes} + 1 \text{ pointer} + 7 \times (1 \text{ bytes} + \text{1 Week Sketches } + m \times g \text{ Event Sketches} ))\
\]
So, with \( P = 64 \) id + i + j + k + l = 5, \( M = 100 \), \( m = 4 \), \( i + j = 2 \), \( E = 4 \), pointer_size = 4 bytes, \( \text{Week Sketches} = 8 \) bits, \( \text{Event Sketches} = 20 \) bits, \( g = 12 \), which are realistic values, our aggregation structure requires 22,01 Mbytes. Such a size is really reasonable considering exchanges, even when dealing with mobile devices.

4. EXCHANGE PROTOCOL

Each driver may decide to publish all or part of the summaries it holds. Of course, s/he can also be interested in all or part of the summaries maintained by others. This selection of the summaries to be exchanged allows reducing bandwidth requirements by avoiding useless transfers. The subscription process consists in defining filters specifying the events types that one is interested in, adding appropriate spatial and temporal criteria. For example, one can be interested in “accidents” occurred in “Paris” during the last month. The information exchanges between vehicles can be done through a relay (e.g. servers located along the roads), or directly between vehicles. The major problem with the exchange is that current wireless technologies are not efficient enough to ensure that the connection between vehicles will not be released before the end of the exchange. One way to handle this problem is to introduce priorities in subscriptions [15] so that cells with a high priority are transferred first.

The exchange process between vehicles \( V_i \) and \( V_j \), where \( V_i \) is the sender and \( V_j \) the receiver, consists of two steps described below:

- **Step 1**: \( V_i \) sends its subscriptions to \( V_j \). \( V_i \) compares \( V_j \)’s subscriptions with its own publications and produces a list of cells (with their associated event and timestamp sketches) to exchange.

- **Step 2**: After the successful exchange of selected cells the couples of sketches \((\text{CP}_i (V_j), \text{CP}_j (V_j))\) selected in **Step 1** are merged using the “inclusive OR” operator, thus ensuring duplicate insensitivity.

Figure 2 illustrates the exchange of summaries between \( V_1 \) and \( V_2 \). \( V_1 \) initiates the exchange with \( V_2 \) at time \( t_1 \). As explained previously \( V_1 \) and \( V_2 \) first exchange their respective priorities. Then, we consider that \( V_2 \) finds a match between its published summaries and \( V_1 \)’s priorities. The types of events required by \( V_1 \) (e.g., accident and available parking space) are also stored on \( V_2 \). Moreover, there is an intersection between \( V_1 \)’s areas of interest \((A_1 \text{ and } A_2)\) and \( V_2 \)’s ones \((A_2 \text{ and } A_4)\). As shown in Figure 2a, \( V_2 \) identifies a single physical cell in common with \( V_1 \) since \( A_1 \cap A_3 = C_9 \). Then, \( V_2 \) identifies the sketches to exchange (i.e., those associated to either accident or available parking place) and corresponding to cell \( C_9 \) (Figure 2b).

At the same time \( V_2 \) compares its priorities with those of \( V_1 \) but there is no match here since they are not interested in same types of event and there is no intersection between \( A_1 \) and \( A_4 \) on \( V_1 \) and \( A_2 \) on \( V_2 \).

In the second and final step, \( V_2 \) sends the selected sketches in a predefined order to \( V_1 \) (e.g. first accident and then available parking place). Then, a merging operation with an “inclusive or” is performed locally on \( V_1 \). The result of this operation is presented in Figure 2c. So, at \( t_1 + \Delta t \) the common physical cell summary on \( V_1 \) changes from Sketches \((V_1, A_2, C_5)\) to Sketches \((V_1, A_2, C_5) + \text{Sketches} \((V_2, A_4, C_5))\).

5. EXPERIMENTAL EVALUATION

5.1 Experimental settings

The VESPA simulator\(^4\), which was used for our experiments, allows simulating realistic urban contexts associated with real cartographic data. Basically, this simulator was developed to evaluate different routing protocols with different traffic conditions and study their impact on the traffic.

To evaluate our aggregation structure and our exchange protocol, we have extended this simulator with modules allowing to build, exchange and exploit event aggregates. The hashing function used for the experiments is SHA-2. In this work, we focused on a single type of events. We chose to evaluate the added-value of our aggregation process on vehicles searching for an available parking space. To simplify the experiments, we did not implement timestamp sketches. We indeed supposed that all vehicles observed events during the same period of time. So, the estimation of the number of event occurrences gives us the estimation of the frequency.

Initially, each simulated vehicle follows the shortest route towards a random target location. When a vehicle leaves a parking place, it broadcasts a message informing the other vehicles about the parking space released. This message is then disseminated among vehicles using the dissemination protocol presented in [3]. All the close-enough vehicles receive it (according to the communication range \( r \) considered of 200 m). Once messages are received on a vehicle, they can be used to change the behavior of the vehicle (e.g., change its direction to drive towards the advertised parking slot), stored in the aggregation data structure or relayed to inform other vehicles. The time needed to send a message from one vehicle to another within its communication range is set to 200 ms.

During the simulations, we considered an environment corresponding to the center of Valenciennes, a city in the north of France. This area was represented by 64 physical cells, where each physical cell was a square of side 300 meters. To evaluate our aggregation process, we placed 8 parking lots around the city. Each parking lot was located in a different physical cell. Moreover, each parking lot has a predefined capacity and a fill rate shown in Table 1. The vehicles located on the parking lots move at 10 kmph whereas the others drive at 30 kmph. Each hour, \( Q \) vehicles \((Q = 100 \text{ in the simulations})\) enter in the city center and start searching for a parking space during 1000 s. If they do not find a free space within this period of time, they will stop searching and continue exchanging data with the other vehicles until they

\(^4\)For more information, see http://www.univ-valenciennes.fr/RGI/SID/tdelot/vespa/simulator.html
exit the simulation (10% of the vehicles entering the simulation leave it each hour). Once a vehicle has found a parking place, it remains parked for a (randomly determined) period of time ranging from 1 hour to 4 hours. Then, the vehicle leaves the place and advertises the released parking spot to its neighbors again.

### 5.2 Criteria and strategies evaluated

In this section, we present the results obtained with different strategies. For each one of them, we studied the evolution of two important criteria when searching for an available parking space: the time needed for each vehicle to find a free space and the percentage of vehicles that actually found a free space within the determined period of time. For vehicles to find an available space, we considered several elementary strategies, and then combined them into more complex ones:

- **View**: this strategy is based on the direct view of an event. Our goal is here to model the classical behavior of a driver searching for an available parking space who is going to park his/her car when s/he sees one.
- **Dissemination**: this strategy considers only the messages diffused by a vehicle leaving its parking space. The vehicles receiving that information are then guided towards this free space.
- **Infrastructure**: this strategy considers the information provided by an infrastructure (i.e., a central server keeping track of all the events occurred) to the drivers. This information consists in a set of reliable statistics about the frequency of all events and the whole concerned area. It is implemented in the simulator as a complete spatio-temporal aggregate data structure
5.3 Qualitative evaluation of the spatio-temporal aggregates

Our first objective was to highlight the interest of the spatio-temporal aggregates for the vehicles searching for an available parking space. Therefore, Figure 3 shows the results produced by the strategies View, Dissemination and Infrastructure concerning the average time needed by vehicles to find a parking space and the percentage of vehicles that actually found a parking space. In Figure 3, we can observe the upper (Infrastructure) and lower bounds (View). We first observe that (whatever the strategy used) the average time for finding an available parking space increases over time. As the same manner, the percentage of vehicle finding a place decreases over time. The reason for these two observations is that the number of vehicles entering into the simulation and searching for an available parking place is higher than the number of free parking slots available (according to the initial parameters defined for the simulations and shown in Table 1). Indeed, the number of parking spaces released on the 8 parking lots is at least 20% less than the number of new vehicles searching for an available parking space. Hence, there is a starvation problem which is getting worst over time and the percentage of cars finding an available space cannot reach 100%. The reason why we chose to show the results for such a congested environment is that the assistance systems should be particularly effective in such configurations where it is very difficult for drivers to find an available parking space.

The results presented in Figure 3 show that the Infrastructure strategy gives significantly better results than the strategies View or View+Dissemination showing the interest of the aggregation data structure. This observation is valid both considering the average time to find a parking space (Figure 3a) and the percentage of vehicles finding a parking space (Figure 3b).

Moreover, View+Dissemination+Infrastructure is the best strategy showing that the corresponding elementary strategies are complementary.

In Figure 4, we introduce the partial aggregation process at the vehicle level (i.e., the SummaryAggregation strategy) and compare it with the strategies studied in Figure 3. Here, we did not consider any exchange of summary between vehicles, the benefits of the exchange protocol will be evaluated later in section 5.4. We analyzed the impact of the range parameter. Please note that the Infrastructure strategy can be considered as the SummaryAggregation with a range of 100% (i.e., the aggregation structure contains all the events occurred nearby). If the results of SummaryAggregation with a range of 50% are compared with the ones obtained for Infrastructure, we notice that even if the "quality" of the data structure is divided by 2, both the average time and the percentage of vehicles finding a resource are not varying in the same proportion. The factor is rather close to 1.5. This is also the case with the ranges 50% and 14% since the results observed are then close to those of the View+Dissemination strategy. This shows that a "perfect" (i.e., complete) aggregation method is not required to benefit from the exploitation of the data structure.

5.4 Evaluation of the exchange protocol

In this section, we evaluate the impact of the exchange of summaries on the quality of the aggregates produced (i.e., the effectiveness of the predictions done with this aggregates). Therefore, we set the range parameter for SummaryAggregation to 50% and compare the results with and without exchanges of aggregates.

Figure 5 shows that performing exchanges between vehicles largely improves the results obtained with SummaryAggregation. The results are then even close to those obtained with the Infrastructure strategy. They show that good results can
be obtained with the SummaryAggregation strategies even with low ranges. Our cooperative scheme so competes with a centralized approach like those considered for the Infrastructure strategy.

6. RELATED WORKS

Aggregation in inter-vehicle networks has so far been considered as a way to optimize storage or to minimize the use of bandwidth. The work presented in [13] describes a system, called TrafficFilter, in which vehicles collaboratively build a speed profile of the road using V2V communications. This system achieves efficient data compression. Instead of averaging information about road segments, only the most relevant single information items for a certain stretch of road are communicated to further away vehicles. To compress vehicle information related to vehicle speeds, Ibrahim and Weigle [6] present a cluster based aggregation scheme suitable for dissemination of vehicle speeds. Contrary to the previous solution, the CASCADE system uses only syntactic, lossless compression of data. At local scope in front of a given vehicle, single reports are disseminated and collected using geo-broadcast. This local view is then clustered using fixed size segments and differential coding is used to compress vehicle information in each cluster. The compressed information is then disseminated further. In [12], RLSMP (Region based Location Service Management Protocol) is proposed. It is based on the aggregation of messages ac-
according to geographical areas. Their goal is to reduce the latest positions and the number of messages generated for the management of vehicle locations.

Works mentioned previously generally consider data summarization as a method of compressing information to save bandwidth. Data compression and data aggregation are distinguished in [11] where the authors present the TrafficView system, which uses semantic aggregation. The authors present two techniques for aggregation: ratio-based and cost-based. In the ratio-based technique, the roadway in front of a vehicle is divided into regions. Data is aggregated based on ratios that have been pre-assigned to each region. Regions farther away from a vehicle are assigned larger aggregation ratios, because precise detail may not be needed over a long range. In the cost-based aggregation technique, data is aggregated based on a cost function that depends on the position of the aggregating vehicle. The produced view of the traffic is not useful to any other vehicle unless it is close to the aggregating vehicle.

Lochert et al. [8] present a probabilistic technique for aggregating the data disseminated in VANETs. The proposed technique does not aggregate the actual values but uses a modified Flajolet-Martin sketch as a probabilistic approximation for the values. This technique can be applied to aggregate the data in any non-accuracy-sensitive application (e.g., estimating the number of available parking spaces), but it cannot be used in our target application, which requires the actual vehicle information to be disseminated and re-aggregated to reach distant vehicles. The interest of the approach in [8] is that it supports any type of event. However, the important memory consumption may be very penalizing. Yu et al. [14] present an aggregation technique called Catch-Up that aggregates similar reports generated by the vehicles whenever an event occurs e.g., a change in vehicle’s density. The technique is based on inserting a delay before forwarding any report in the hopes of receiving similar reports from surrounding vehicles so that these reports can be aggregated into a single report. Lochert et al. [9] describe a hierarchical aggregation technique for vehicle travel times. In this technique each vehicle broadcasts its travel time between two landmarks along its trip. Then these travel times are aggregated hierarchically and broadcasted to provide distant vehicles with an estimate of the travel times along the road segments so that they can avoid congested roads (the roads with larger travel time estimate). As with Catch-Up, this work does not disseminate or aggregate information suitable for safety applications, but is only concerned with reporting traffic conditions.

7. CONCLUSIONS

In this article, we have presented a cooperative scheme to aggregate events in vehicular networks in order to produce useful information for the drivers. This scheme is based on a spatio-temporal aggregation data structure deployed on the vehicles. This data structure focuses on spatio-temporal accesses and is designed to be (partly) exchanged without loss of information between vehicles. Our aggregation structure is realistic in size if the number of temporal dimensions remains controlled. The complexity to access the structure is also efficient (logarithmic or linear). Our proposal has also been evaluated and validated by a set of experiments made with an extension of the VESPA simulator. The results obtained show that our aggregation data structure provides good results. The use of our structure indeed reduces the time needed to find a parking space and increases the percentage of vehicles actually finding a place.

We are currently studying more complex strategies to exploit the aggregation data structure, for instance not to restrict the search of the best area to the cells at a distance of 1 from the ones the user is located in. Moreover, in order to
improve the percentage of information exchanged between vehicles, we are currently working on prediction techniques to optimize the use of the connection time between two vehicles willing to exchange (parts of) their summaries.

8. REFERENCES


