Building, sharing and exploiting spatio-temporal aggregates in vehicular networks

Dorsaf Zekri[1,2], Bruno Defude[1] and Thierry Delot[2,3*]

[1]Institut TELECOM, TELECOM SudParis
UMR CNRS SAMOVAR, Evry, France

[2]University of Valenciennes
LAMIH UMR CNRS/UVHC 8201, Valenciennes, France

[3]Inria Lille – Nord Europe, Lille, France

July 24, 2013

Abstract

This article focuses on data aggregation in vehicular ad hoc networks (VANETs). In such networks, data produced by sensors or crowdsourcers are exchanged between vehicles in order to warn or inform drivers when an event occurs (e.g., an accident, a traffic congestion, a parking space released, a vehicle with non-functioning brake lights, etc.). In the following, we propose to generate spatio-temporal aggregates containing these data in order to keep a summary of past events. We therefore use Flajolet-Martin sketches. Our goal is then to exploit these aggregates to better assist the drivers. These aggregates may indeed produce additional knowledge that may be useful when no event has been recently transmitted by surrounding vehicles or when some knowledge about the global demand may improve the decision that need to be taken at the vehicle level.

To prove the effectiveness of our approach, an extensive experimental evaluation has been performed considering vehicles looking for an available parking space, that proves the interest of our proposal. The experimentations indeed show that the use of our aggregation structure significantly reduces the time needed to actually find a parking space. It also increases the percentage of vehicles finding such a resource in a bounded time in congested situations.

Keywords: Vehicular ad hoc networks, information sharing, spatio-temporal data aggregation, Flajolet-Martin sketches, communication protocols.

[*]Corresponding author: Thierry Delot, University of Valenciennes, LAMIH UMR CNRS 8201 and Inria Lille – Nord Europe, Le Mont Houy, 59313 Valenciennes Cedex 9, France ; Tel: +33 3 27 51 19 56 ; Fax: +33 3 27 51 19 40 ; e-mail: Thierry.Delot@univ-valenciennes.fr.

1 Introduction

Nowadays, there is a great interest in developing systems to assist drivers on the road, providing them with different types of relevant information. VANETs rely on the use of short-range networks (a few hundred meters), like IEEE 802.11, Ultra Wide Band (UWB), or WAVE (IEEE 802.11p, IEEE 1609), for vehicles to communicate [19] and provide bandwidth in the range of Mbps. VANETs allow vehicles to cooperate so that drivers can be informed that an accident has occurred or that a traffic congestion has appeared on the road a few hundred kilometers ahead [26].

The work described in this article takes place in the VESPA project [10], a system designed for vehicles to share information in inter-vehicle ad-hoc networks [11]. The main originality of VESPA is to support the exchange of any type of event in the network (e.g., available parking spaces, accidents, emergency braking, obstacles in the road, real-time traffic information, information relative to the coordination of vehicles in emergency situations, etc.). Therefore, VESPA proposes a dissemination protocol based on the concept of Encounter Probability to estimate the relevance of events for vehicles [6].

Data aggregation is defined by [21] as a technique used to overcome two problems: *implosion* (i.e., data sensed by one node is duplicated in the network due to data routing strategy) and *overlap* (i.e., two different nodes disseminate the same data). Recently, data aggregation has thus been exploited by many of these communication protocols designed for vehicular networks [8, 31, 28, 22, 4]. However, data aggregation techniques are only considered in this context as a method for compressing data in order to reduce bandwidth usage. The approach described in this article is quite different. Our goal is indeed to summarize information about previously observed events and then to extract from produced aggregates useful environmental knowledge for the drivers. For instance, a summary of parking spaces recently released may be helpful to identify the area with a high probability of finding free places at a given day and hour. In a different context, thanks to the correlation of safety related messages received by a vehicle (e.g., an accident, an emergency braking, etc.), dangerous areas can be dynamically detected and indicated to the driver. Such an approach can be applied not only to the detection of permanently dangerous areas but also to the temporarily ones due to bad weather conditions for example.

Obviously, each vehicle has only a limited view since it can not observe or receive the notifications about all occurring events. Therefore, we also introduce protocols for vehicles to share (parts of) their aggregates and thus improve their knowledge base.

The main originality of our approach resides in its capacity to exploit *deprecated* information to predict the future what is clearly unusual. Indeed, in VESPA as well as in the other existing systems, messages representing events (e.g., a traffic congestion, an emergency braking, a parking space released, etc.) are disseminated using various protocols in order to warn or inform drivers. However, data is considered as an “object” to transmit and deleted once used. On the contrary, we consider with a data management point of view that this information can still be useful to assist drivers. It can indeed be exploited to guide the drivers when no information has recently been provided by neighboring drivers or to achieve the best choice among several alternatives (e.g., determine the best target considering the global demand when several available parking spaces have been notified to a same vehicle) Several challenges have to be overcome to define and implement such summaries. First of all, the aggregation process has to deal with duplicate events (i.e., the same event received by several vehicles). The cost of communications is another issue. Vehicles can communicate indeed communicate directly with each other or through an infrastructure. However, the bandwidth is quite limited and the connection time too (e.g., up to a few seconds regarding inter-vehicle communications). Obviously, it is also possible to exploit cellular networks but coverage, privacy and scalability issues remain. Thus, for the moment existing solutions are limited to the scale of a city for specific types of events). Collaborative solutions where vehicles construct their own summaries and exchange them with other vehicles are more adequate, but need frequent exchanges. Finally, determining a good tradeoff between the size and the accuracy of the summaries is also quite challenging.

Summing up, the main contributions of this paper are the following:

- *We propose a general aggregation structure for vehicular networks.* This data structure integrates both spatial and temporal dimensions to aggregate events and requires a limited storage space.
- *We propose an exchange protocol for vehicles to share (parts of) their respective aggregates.* By supporting preferences about drivers' interests, this protocol can cope with the constraints imposed by vehicular networks on the exchange (e.g., short connection times, low bandwidth, etc.). Moreover, the characteristics of our data structure allow easily merging the fragments received with the original aggregate hold by the vehicle.
- *We perform an extensive experimental evaluation to test and validate the efficiency of the aggregation data structure and the exchange protocol.* The experimental results show the interest of the approach through different use cases.

The rest of this article is organized as follows. In section 2, we present the global approach considered in this work and introduce some preliminary concepts. Section 3 introduces Flajolet-Martin sketches and describes the proposed aggregation structure. Section 4 focuses on how aggregates built on distinct vehicles can be exchanged and merged. In section 5, we show the effectiveness of our solution through various experimental results. In section 6, we compare our approach with related works. Finally, we conclude and present the perspectives of our work in section 7.

2 General context

In the following, we consider smart vehicles able to provide alert services and decision support to drivers. Thus, as described in Figure 1, a vehicle i may acquire information about events observed either by itself (e.g., via embedded sensors for example) or diffused by other vehicles (e.g., using a dissemination protocol). In this case, the information available on each vehicle is partial and incomplete since vehicles cannot perceive all occurring events or receive all messages transmitted by other vehicles using short-range wireless communications. To complete their local information, vehicles may also sometime acquire information from a fixed infrastructure deployed along roads. For example, in urban areas, the infrastructure may correspond to a central parking management system or a central traffic information server providing information to vehicles driving in its vicinity.

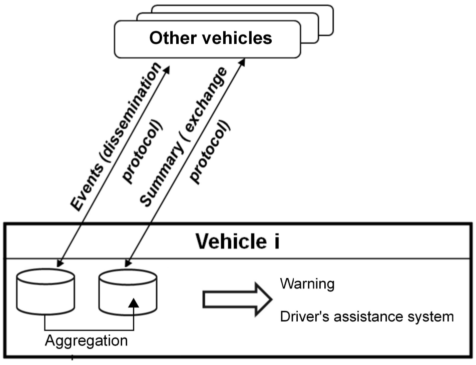

Other vehicles
Events (dissemination protocol)
Summary (exchange protocol)
Vehicle i
Warning
Driver's assistance system
Aggregation

Figure 1: Global architecture

Usually, events broadcasted in the vehicular network have a quite short lifetime, ranging from a few seconds (e.g., an emergency braking) to several hours (e.g., a traffic congestion) depending on their type. Table 1 represents a simple message created to advertise an available parking space. In this example, the message contains a unique identifier, a priority (e.g., to make sure that safety related messages will be treated before comfort ones), the reference position of the physical event (e.g., the GPS coordinates of the available parking space) and the type of the event considered. Thanks to one of the existing dissemination protocols [37, 5, 25], this message is then transmitted to the vehicles driving in the vicinity of the parking place during a limited period of time.

The solution presented in this article does not depend on any specific protocol used to broadcast information in the network. We only assume that vehicles receive messages containing at least the attributes depicted in Table 1. In terms of communication features, we consider that vehicles support at least short-range communications (e.g., Wi-Fi or DSRC).

Identifier	**Priority**	**Position & Date**	**Description**
20130303102517501915917$N$305111$EAD$	*low*	50°19'15.91 N 3°30'51.11 E 10h25m17s 2013-03-03	Parking released

Table 1: Example of message representing an available parking space

To avoid losing information related to the events observed (e.g., the available parking spaces), we propose in this article to aggregate events considered obsolete (i.e., previously observed and possibly used to produce a warning to the driver) at the vehicle level. To improve their quality and coverage, summaries are also exchanged between vehicles using an exchange protocol. These summaries are then used to estimate the probability that an event can happen, even without any real-time observation. Thus, when many accidents are observed in a particular geographical area, it is possible to conclude that this area is dangerous enough to warn drivers, even if no accident has been signaled by a neighboring at this time.

An alarm management module or an assistance system for drivers can benefit from events observed by the vehicle (or others), from information delivered by an infrastructure and from summaries built on the vehicles (or exchanged with others). For example summaries can be used to recommend areas where the probability to get free parking places is high. Different strategies can be applied to compute such recommendations depending for example on the size of the recommended areas. Obviously, the confidence in the information is also an important parameter which may change since the summaries do not contain real but probabilistic information. For instance, the enhancement/reduction of the confidence value affected to a summary may depend on the drivers' feedback.

3 Aggregation structure

The definition of the summarization process in our work is "to aggregate past events to provide a knowledge base to estimate whether an event might occur even without observation". Obviously, a variety of techniques exist that can be used to build summaries of spatio-temporal events. In our case, the important criteria expected for a summary are:

- To estimate the frequency of (type of) event occurrences;
- To promote basic dimensions that are location and time;
- To be incrementally constructible and inexpensive in both computing time and storage space;
- To let each driver define the types of events s/he is interesting in, as well as the spatial and temporal scales s/he wants to consider in the aggregation process;
- To allow exchanging and merging (parts of) summaries between vehicles so that they can enrich their respective knowledge base.

The first criterion requires a compact representation. The last criterion implies that the aggregation mechanism detects duplicates. Therefore, it has to recognize when the same event has been observed by two different vehicles in order not to consider it as two different events to aggregate.

To achieve these objectives, we rely on the two-level spatio-temporal model presented in section 3.1. We also exploit Flajolet-Martin sketches introduced in section 3.2. Then, we detail our aggregation data structure in section 3.3 and propose a theoretical evaluation in section 3.4.

3.1 Two-level spatio-temporal model

To address these needs expressed previously, we first propose the two-level model illustrated in Figure 2. This model is composed of two main parts:

- The *physical level* constitutes the lowest level. It consists in a repository shared between all vehicles which goal is to allow information exchanges without loss. The physical level is divided into fixed size squares that form a full partition. The same idea is used for the temporal dimension. Time is so divided into segments that form a full partition. We assume here that we want to emphasize the seasonal nature of the event production. We therefore propose to split the time in 7 days, themselves sliced in 2 hours segments, providing a total of 84 time segments. The couple {*square*, *time segment*} is the smallest unit that can count occurrences of events. This physical space is very large: assuming that the size of a cell space is 1 km^2 and 10 time segments, the coverage of France would represent about 6 million pairs. This number could be reduced by structuring the space using unfixed size areas, which allow having a better spatial resolution in urban areas (and greater accuracy). However, this requires a little bit more complex algorithm to implement.

- The *logical level* allows each driver to define her/his preferences. Based on this physical level, a specific logical splitting can be specified on every vehicle and defined as a set of rectangles (or intervals). Those rectangles represented with dashed lines in Figure 2 are themselves composed of squares (or intervals) of the physical layer. Indeed, a driver may not be interested in monitoring the whole space but only in a subset of spatio-temporal segments. The choice of the logical cells can thus be specified by the driver. It may also be customized according to the driver's displacements by observing frequently visited areas. The number of squares (intervals) actually observed at the logical level is so (much) smaller than the whole physical level. For example, if a driver wants to monitor one hundred of spatial areas covering an average surface of $20km^2$, the number of couples to consider is approximately equal to 2000.

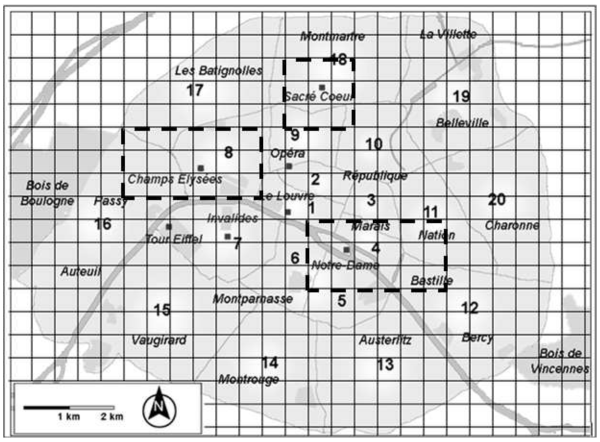


Figure 2: Two-level spatial model

At this stage, one may wonder why we distinguish the physical and logical levels. The main reason is related to the need of merging exchanged information.

Indeed, since all the cars only have a partial view of the events generated, exchanges of aggregates are needed to increase the content of the knowledge base and the quality of the indications delivered to the driver. We therefore propose an exchange protocol that will be presented in detail in section 4. In the following, we just illustrate the interest of creating the interest areas on top of our physical model to avoid loosing information through a simple example. Let us consider the exchange of information gathered by two vehicles with their own division of the spatio-temporal space as described in Figure 3. In this example, *Vehicle 1* holds 3 interest areas represented by a cell determined by the coordinates of the bottom left and upper right corners. Each cell contains the aggregate value (e.g., the number of events observed in this area). As the same manner, events are aggregated for 4 different interest areas on *Vehicle 2*. The unique intersection between the respective interest areas of *Vehicle 1* and *Vehicle 2* are represented with dashed-lines on *Vehicle 1*.

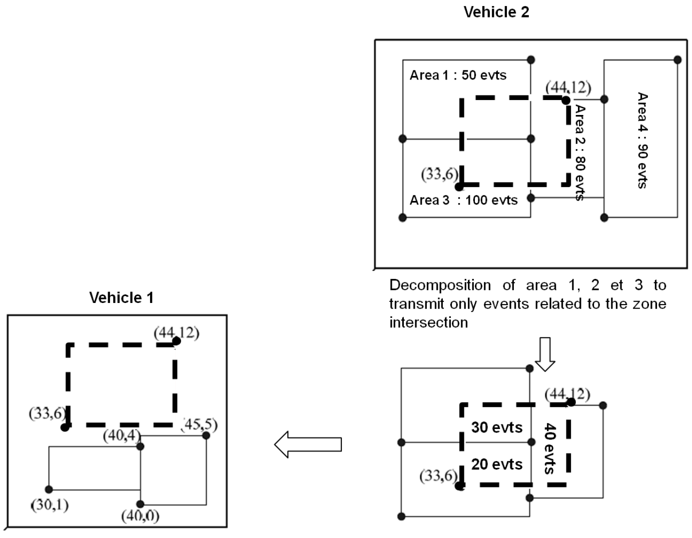


Figure 3: Exchange of summary between two vehicles

Both to optimize the volume of data transferred and to actually merge the (parts of) aggregates exchanged with the one already maintained on a vehicle, the intersection between the respective interest areas has to be computed. Therefore, interest areas may have to be split into sub-cells since each vehicle has its own division. For instance, *Area 1* on *Vehicle 2* has to be "divided" into two sub-areas to perform the merging phase since it only partly matches with the interest area of *Vehicle 1*. Since the accurate positions of the events (e.g., their GPS coordinates) have been lost during the aggregation process, it is not possible to precisely allocate the 50 events contained in *Area 1* over both generated sub-areas any more. Hence, the number of events initially observed have to be distributed between the different fragments (considering a uniform distribution for example). Obviously, such an approximation leads to an increase of the imprecision and impact the quality of the predictions performed using the aggregates. On the contrary, our choice to impose the same physical model on top of which drivers may define their areas of interest easily avoids these problems.

In this sub-section, we have introduced our two-level spatio-temporal model. In the following one, we introduce Flajolet-Martin sketches exploited in this model to aggregate events.

3.2 Sketches

Flajolet-Martin Sketches [16] provide a compact representation to estimate the number of occurrences of distinct objects. The sketch contains a set of binary arrays initially filled with 0. The size of the sketch is defined according to the size of the array. The longer the chain, the better the accuracy of the estimate. The insertion of an object into a Flajolet-Martin sketch is represented in Figure 4. A hash function h is first applied to the element x to insert. Let $lwp(h(x))$ be the position of the rightmost value 1 in the binary representation of x. The bit with index $lwp(h(x))$ is then set to 1 in the sketch if its value was still 0.

Once the sketch has been constructed, the number of distinct values p contained is the sketch can be estimated using the estimate function *E(p)*=$log_2(\phi n)$, where n is the position of the leftmost 0 in the binary table and ϕ is a correction factor [16].

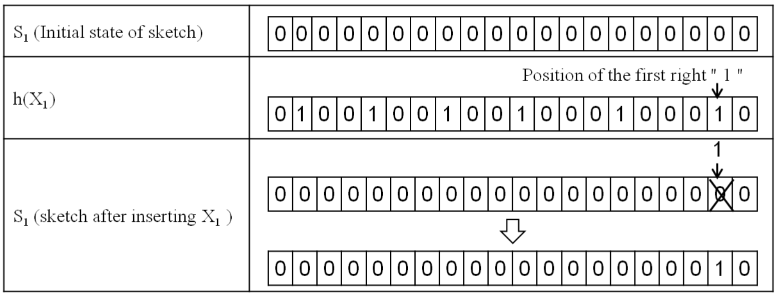


Figure 4: Insertion of an event in a Flajolet-Martin sketch

Due to the exchange and merging constraints, we cannot use simple counters to aggregate information as proposed in [9]. This would indeed lead to a loss of information at the merging stage illustrated in the previous example. Flajolet-Martin sketches thus provide an interesting alternative since they detect duplicates by construction. Two instances of the same event indeed have the same image computed by the hash function.

This sketch has been used in [34] which proposes a method for spatio-temporal indexing based on a R-tree for the spatial part and a B-tree for the time part. The value stored in a tree cell is a sketch and not a simple integer. This method is not suited for performing exchanges without loosing information. It indeed uses static regions which can be divided into sub-areas causing loss of information.

3.3 Aggregation data structure

In this the following, we describe the aggregation data structure we propose to implement the spatio-temporal model introduced in section 3.1. In this work, we assume that each vehicle V_i can observe a set of events E. Each event e of E is characterized by[1]:

1. ty_e: the type of observed event (e.g., accident, released parking space, etc.).

2. lo_e: the location of the event and its timestamp. This information is provided by GPS like positioning systems.

3. id_e: the unique event identifier of e. This unique identifier is the basis for the detection of duplicates. We assume that an instance of event always produces the same identifier on vehicle V_i and on other vehicles. Such a unique identifier can be generated by combining the current time and the GPS location of the event with a randomly-generated sequence[2].

According to the spatio-temporal model previously introduced, we assume that the physical space is divided into C_{NP} squares (N squares on the X axis and P ones on the Y axis) with g temporal granularities. The coordinates of the origin point are (x_{origin}, y_{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* than at the *physical level*. For example, we can use 84 temporal granularities ordered from g_1 (Monday from 12.00 am to 2.00 am) to g_{84} (Sunday from 10.00 pm to 12.00 am).

The data structure supporting this spatio-temporal organization is illustrated in Figure 5. This aggregation data structure allows a quick access to the interest areas according to spatial coordinates. It consists of an ordered array of interest areas. Each interest area is given a unique identifier *id* and is bounded by two physical cells (e.g., the bottom left and upper right ones). In our example, 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 *id*. Moreover, each interest area is associated with a linked list representing the physical cells it contains (6 cells in our example). For each of these cells, we finally store g items (according to the defined temporal granularity) containing the frequency of observed events for the corresponding *physical cell* for a given time granularity. We estimate this frequency of events as the ratio between the number of observed events and the number of observed weeks. For example, if 200 events are observed during 4 weeks, the frequency is set to 50. Therefore, we use both *event sketches* to estimate the number of events observed and *timestamp sketches* to estimate the number of observed weeks. The event sketch is constructed by applying a hash function on the identifier id_e of each event observed. The timestamp sketch is constructed by applying a hash function on the number of the week.

[1] These items are only considered in the summary, but other information can be useful for managing alarms or disseminating messages in the network.

[2] 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 problem. Interesting ideas to solve it have been proposed in the field of information fusion [15, 20].

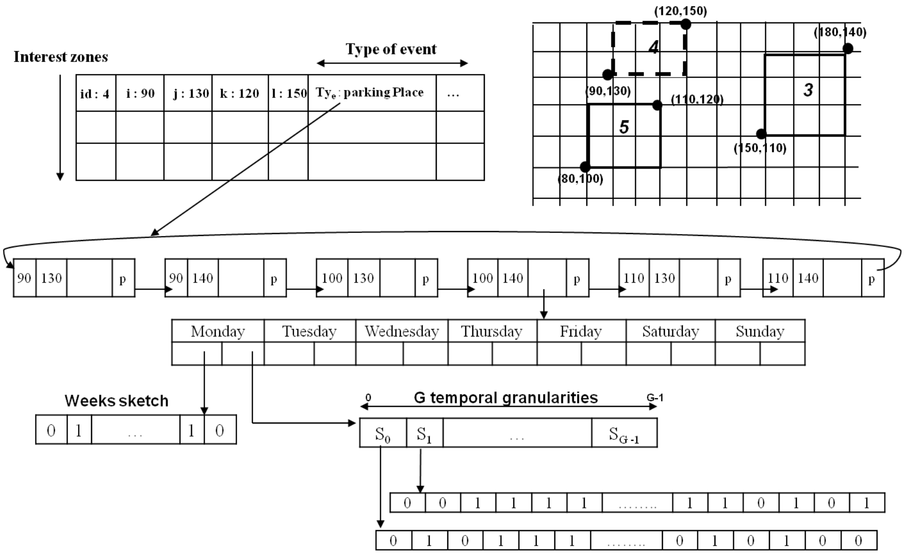


Figure 5: Spatio-Temporal Aggregation Data structure

When all objects are stored, the number of distinct objects is 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) [16]. 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, we apply the hash function 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 obtain a good precision.

3.4 Theoretical evaluation

In this section, we provide some elements to appreciate the effectiveness of our data structure. Obviously, a detailed experimental evaluation has been conducted which results prove the interest of the approach. However, we want at this stage to provide some elements showing that our data structure matches the expectations in terms of both storage space and access cost.

In our aggregation data structure, the size of a sketch depends on the maximum number of items it should contain. Here, we assume that we may have to store up to 1.000.000 events in a cell. Therefore, 20 bits are required for each event sketch. Moreover, 8 bits are needed per timestamp sketch to monitor the events over 256 weeks. 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 over the whole week (7 days). The storage space required for the aggregation structure can thus be computed by:

$Storage\ space = P \times [((id + i + j + k + l)\ bytes + E\ bits + E\ pointers) +$

E × *M((i + j) bytes + 1 pointer + 7* × *(1 bytes + 1 Week_Sketches + m* × *g Event_Sketches))]*

With $P = 64$*, id + i + j + k + l = 5 bytes,* $M = 100$*,* $m = 4$*,* $i + j = 2$*,* $E = 4$*, pointer_size = 4 bytes, Week_Sketches = 8 bits, Event_Sketches = 20 bits,* $g = 12$, which are realistic values, the storage space required for our aggregation structure only reaches 22,01 Mbytes. This shows the compactness of our structure which can thus even be used on mobile devices embedded in the cars.

Concerning the access cost to a specific physical cell of the data structure, it is linear in the number of areas and in type of event: $O(P + E \times M)$.

In this section, we have introduced an aggregation data structure exploiting Flajolet-Martin sketches for vehicles to summarize information about observed events. In the following, we focus on the (partial) exchange of summaries built on the vehicles in order to enrich the local database of each vehicle that can be used to extract information for the driver.

4 Exchange Protocol

4.1 General principle

Each vehicle may publish all or part of its summaries to other vehicles. Each vehicle may also be interested in all or part of the others' summaries. To simplify, we consider in the following only public publications and subscriptions (i.e., one publishes/subscribes to all the vehicles it is likely to meet). The publication process consists in defining which summaries should be published (possibly aggregating them by grouping cells).

The subscription process consists in defining filters specifying the events types that the driver is interested about, adding appropriate spatial and temporal criteria. For example, a driver can be interested by Accidents in "Paris" over the last month. The exchange of information between vehicles can then be done through a relay (e.g., servers located along the roads), or directly. In both cases, the exchange process is unsure if the duration of the connection is not sufficient to allow the complete exchange of summaries. We therefore propose to use a mechanism based on priorities, which defines an order based on data utility, and use this order to prioritize exchanges. Priorities are defined as a set of rules defining an order between several elements. We use as elements the various types of events, different time granularities for the temporal dimension and the different areas of interest to address the spatial dimension.

The following example illustrates the exchange priorities of vehicle V_i. The following expression describes the types of events V_i is interested about (i.e., Accident first, then Available Parking Space). Implicitly, all the other types are here considered non relevant:

(Exp 1) Accident > Available Parking Space

Priorities may be expressed in the same way over the temporal granularities ranging for example from g_1 (Monday - 12.00 am to 2.00 am) to g_{84} (Sunday -

10.00 pm to 12.00 am).

Similarly, if we assume 10 areas of interest for V_i, the next expression defines an order between them:

(Exp 3) $A_1 > A_3$ *;* $A_2 > A_4$ *;* $A_4 > A_6$ *;* $A_6 > A_8$

In this case, we have a partial order with A_1 and A_2 which are prior areas, then A_3 and A_4 then A_6 and finally A_8. Non-mentioned areas are not affected by the exchange. $Exp1$, $Exp2$ and $Exp3$ define the priorities to follow when vehicle V_i receives data from another vehicle. Again, these priorities may be set by the drivers or adapted dynamically according to the drivers' displacements. In this last case, drivers' destinations can be exploited to determine the areas for which data should be gathered (e.g., because the driver is visiting it regularly or because no information is currently available on the vehicle for these areas).

When V_i meets V_j and if needs to obtain new summaries, it starts sending information about its priorities. Then, V_j calculates the intersection among its summaries and the received priorities. If that intersection is not empty, it sends data corresponding to requested priorities. Depending on the duration of the connection between vehicles, all or part of the exchange will be realized. Obviously, exchanges should rather be initiated when vehicles are stationary (e.g., stopped at a traffic light) or driving at low speed.

The basic operation here is to compute the intersection between two interest areas (i.e., the one of each vehicle). This intersection returns either the empty set (i.e., the two areas are distinct) or a set of physical cells when they have an intersection. For these common cells, the result is just the "inclusive OR" of sketches. The cost of this calculation is logarithmic in the number of areas (to determine the p intersecting areas) and linear in number of physical cells: $O(logP + p \times M)$.

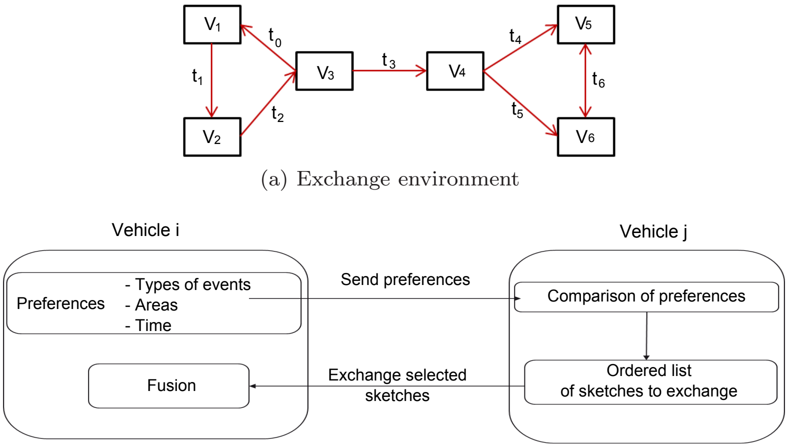
Obviously, V_i should not exchange continuously with the same vehicles. Therefore, each vehicle stores a list containing the identifiers of N latest vehicles with which an exchange took place as well as their timestamp. Before initiating the exchange with a vehicle, the system has so to verify that the identifier of the encountered vehicle does not already appear in this list.

Another problem to avoid in the exchange phase is the one of duplicates (i.e., counting several times the same event occurrences). This problem is solved using Flajolet-Martin sketches and applying a hash function to the key of the events. Indeed, if two vehicles V_i and V_j observe the same occurrence of an event idf_e, the same hash function h is applied on both vehicles. Thus, $h_{V_i}\ (idf_e) = h_{V_j}\ (idf_e)$ and the use of the "inclusive OR" only retains one occurrence in the exchange of sketches.

The exchange process is summarized in Figure 6. In Figure 6a, we consider 6 vehicles close enough to communicate, knowing that the exchange will take place successively at times ($t_0 < t_1 < t_2 < ... < t_6$). A directed edge between two vehicles V_i and V_j means that V_i's summary has been updated thanks to V_j's sketches. Let us concentrate in the following on the exchange between vehicles V_1 and V_2, where V_1 is the sender and V_2 the receiver. The exchange is composed by two main steps described in the following and illustrated in Figure 6b:

- *Step 1*: V_1 sends its preferences to V_2. V_2 compares V_1's priorities with its own sketches. According to the matches, V_2 produces an ordered list of sketches to exchange. This step notably implies to transform the partial order defined by priorities in a total one. Therefore, we give priority for the space dimension to the areas which are close to the current one. As the same manner, we favor the most recent ones for the time dimension.

- *Step 2*: This phase consists in the actual exchange of sketches selected according to the order computed at *Step 1*. Then, the set of sketches sent to V_1 by V_2 are merged the ones previously hold by computing an "inclusive OR" between both sketches. If the connection time is sufficiently important, all selected sketches are actually exchanged. Otherwise, only the preferred sketches are exchanged and merged on the recipient vehicle.



(a) Exchange environment

(b) Exchange process

Figure 6: Exchange principle

4.2 Example of exchange between two vehicles

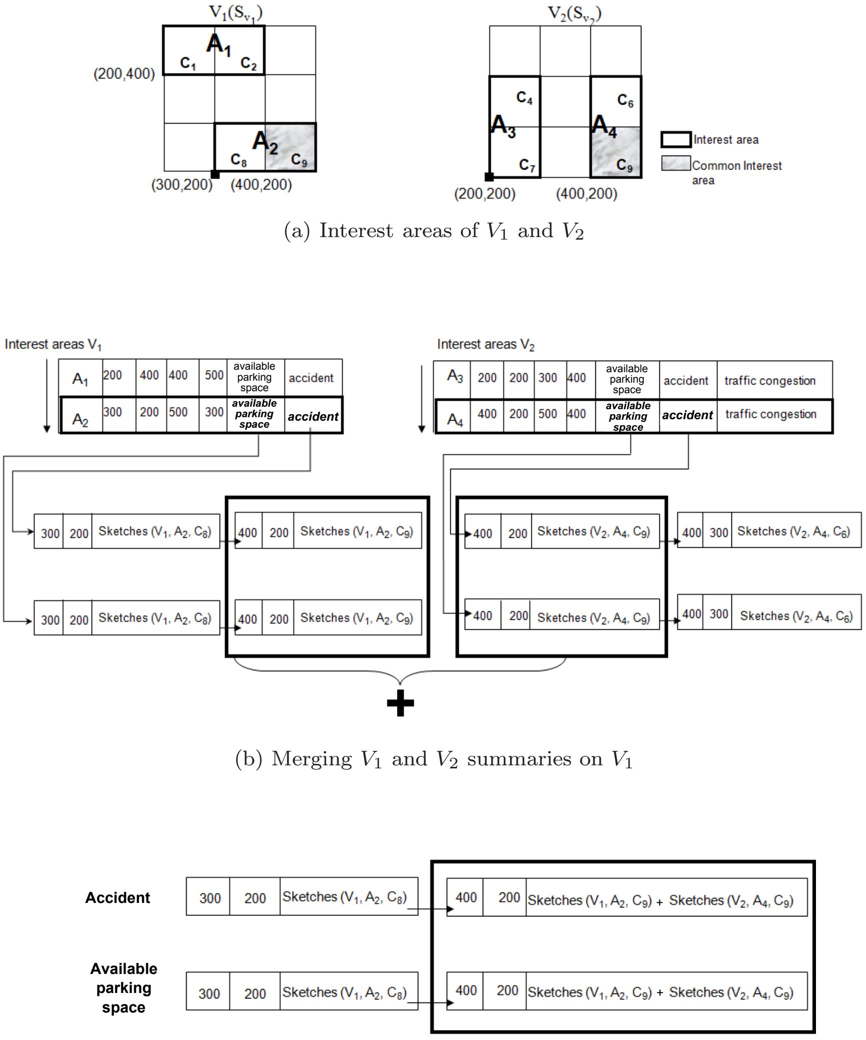
In this section, we present the details of the exchange of sketches between V_1 and V_2 occurring at step t_1. This exchange is represented in Figure 7. In our example, we assume that:

- V_1 holds a summary S_1 and is interested in two types of event (Accident and Available Parking Space) for two logical areas A_1 and A_2. Each of these areas is composed by two physical cells, respectively c_1, c_2 for A_1 and c_8, c_9 for A_2.
- V_2 holds a summary S_2 and is interested in three different types of event (Accident, Available Parking Space and Traffic Congestion) for two logical areas A_3 and A_4. Each of these logical areas is composed by two physical cells: (c_4, c_7) for A_3 and (c_6, c_9) A_4 as depicted in Figures 7a and 7b.

These vehicles also have priorities defined about the types of event, the spatial zone (interest area) and the time frame (temporal granularity) they need to monitor. These priorities for V_1 and V_2 are expressed as follows:

$$V_1's\ priorities \begin{cases} Accident > Available\ Parking\ Space \\ g_1 > g_2 > ... > g_{12} \\ A_2 > A_1 \end{cases}$$

$$V_2's\ priorities \begin{cases} Traffic\ Congestion \\ g_1 > g_2 > ... > g_{12} \\ A_3 \end{cases}$$



$V_1(S_{v_1})$
A_1
C_1
C_2
(200,400)
A_2
C_8
C_9
(300,200)
(400,200)
$V_2(S_{v_2})$
C_4
C_6
A_3
A_4
C_7
C_9
(200,200)
(400,200)
Interest area
Common Interest area
Interest areas V_1
A_1 200 400 400 500 available parking space accident
A_2 300 200 500 300 available parking space accident
Interest areas V_2
A_3 200 200 300 400 available parking space accident traffic congestion
A_4 400 200 500 400 available parking space accident traffic congestion
300 200 Sketches (V_1, A_2, C_8)
400 200 Sketches (V_1, A_2, C_9)
400 200 Sketches (V_2, A_4, C_9)
400 300 Sketches (V_2, A_4, C_6)
300 200 Sketches (V_1, A_2, C_8)
400 200 Sketches (V_1, A_2, C_9)
400 200 Sketches (V_2, A_4, C_9)
400 300 Sketches (V_2, A_4, C_6)
+
Accident
300 200 Sketches (V_1, A_2, C_8)
400 200 Sketches (V_1, A_2, C_9) + Sketches (V_2, A_4, C_9)
Available parking space
300 200 Sketches (V_1, A_2, C_8)
400 200 Sketches (V_1, A_2, C_9) + Sketches (V_2, A_4, C_9)

(a) Interest areas of V_1 and V_2

(b) Merging V_1 and V_2 summaries on V_1

(c) Updated sketches on V_1

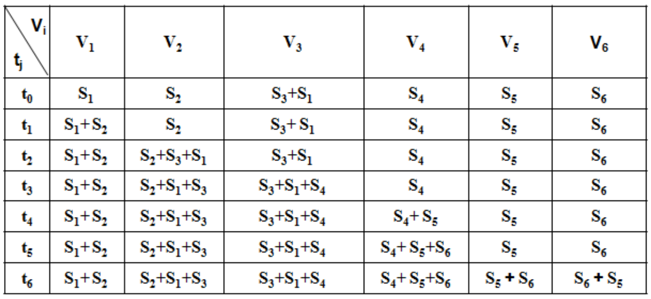
Figure 7: Exchange of sketches between vehicles V_1 and V_2

As shown in Figure 6a, V_1 initiates the exchange of summaries with V_2 at time t_1. At the first step, V_1 and V_2 exchange their respective priorities. Then,

V_2 finds a match between its summaries and V_1's priorities. The temporal granularities are indeed the same on both vehicles. Moreover, the types of events required by V_1 (e.g., Accident and Available Parking Space) are stored on V_2. Finally, there is an intersection between V_1's areas of interest (A_1 and A_2) and V_2's ones (A_2 and A_4). As shown in Figure 7a, V_2 identifies a single physical cell in common with V_1 since $A_1 \bigcap A_2 = C_9$. Hence, V_2 identifies the sketches to exchange (i.e., those associated to either Accident or Available Parking Space for all time periods) and corresponding to cell C_9 (Figure 7b). 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_2 on V_1 and A_2 on V_2.

In the second and final step, V_2 sends the selected sketches in the defined order to V_1 (e.g. first Accident and then Available Parking Space). Then, a merging operation with an "inclusive OR" is performed locally on V_1. The result of this operation is presented in Figure 7c. At $t_1 + \triangle t$ the summary associated to physical cell C_9 on V_1 changes from Sketches (V_1, A_2, C_9) to Sketches (V_1, A_2, C_9) + Sketches (V_2, A_4, C_9).

To generalize, we represent the sequence of summaries' exchanges in Figure 8. In this table, a cell *(i, j)* contains the value summarized on vehicle V_i at time t_j. This illustrates that exchanges improve the completeness of vehicles' summary. For instance, V_4 improves its initial summary S_4 by merging the values of S_4, S_5 and S_6 so changes from S_4 to S_4+S_5+S_6 at t_5 as shown in Figure 8. Let us note also that the exchange process can be bidirectional provided that the connection time between vehicles is long enough. This is illustrated by the two edges between V_5 and V_6 at t_6 in Figure 6a.



V_i / t_j	V_1	V_2	V_3	V_4	V_5	V_6
t_0	S_1	S_2	S_3+S_1	S_4	S_5	S_6
t_1	S_1+S_2	S_2	S_3+ S_1	S_4	S_5	S_6
t_2	S_1+S_2	S_2+S_3+S_1	S_3+S_1	S_4	S_5	S_6
t_3	S_1+S_2	S_2+S_1+S_3	S_3+S_1+S_4	S_4	S_5	S_6
t_4	S_1+S_2	S_2+S_1+S_3	S_3+S_1+S_4	S_4+ S_5	S_5	S_6
t_5	S_1+S_2	S_2+S_1+S_3	S_3+S_1+S_4	S_4+ S_5+S_6	S_5	S_6
t_6	S_1+S_2	S_2+S_1+S_3	S_3+S_1+S_4	S_4+ S_5+S_6	S_5 + S_6	S_6 + S_5

Figure 8: Sequence of summary's exchanges

5 Experimental Evaluation

In this section, we present some experimental results to show the effectiveness of our aggregation structure and the impact of the exchange protocol. During the experimentations, we decided to focus on vehicles searching for an available parking space. Using vehicular communications to facilitate such search has recently become a popular problem in the mobile data management community [12, 2, 36]. Moreover, this use case provides us effective measures to actually evaluate the efficiency of our data structure and its associated exchange protocol (e.g., the average time for vehicles to find an available parking spot or the percentage of vehicles actually finding a free parking spot in case of starvation).

5.1 Experimental settings

The VESPA simulator[3], which was used for our experimentations, allows simulating realistic urban contexts associated with real cartographic data. Basically, this simulator was developed to evaluate different routing protocols with different traffic conditions, such as [11, 13], 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 aggregates. The hashing function used for the experiments is SHA-2 [17]. In this work, we focused on a single type of event. More precisely, we chose to evaluate the added-value of our aggregation structure on vehicles searching for an available parking space.

Initially, each simulated vehicle follows the shortest path towards a random target location. When a vehicle leaves a parking place, it broadcasts a message informing close vehicles about the parking space released. This message is then disseminated among vehicles using the dissemination protocol presented in [10]. Each time a message is sent by a vehicle, all close-enough vehicles receive it (according to the considered communication range r of 200 m). Once a message is received by a vehicle, it can either 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, relayed to inform other vehicles or discarded. Finally, the time needed to send a message from one vehicle to another within its communication range was set to 200 ms during our experimentations.

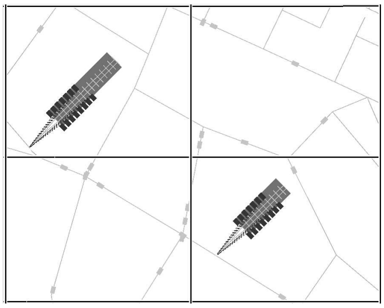


Figure 9: Graphical interface of the simulator

During the simulations, we monitored an area corresponding to the center of Valenciennes, a city located in the north of France. This area was represented by 64 physical cells. Each physical cell was a square of side 200 meters. A snapshot of the simulator's interface is presented in Figure 9.

To evaluate our aggregation process, we placed 8 parking lots around the city. Each parking lot was located in a different physical cell. Each parking lot had a predefined capacity and a fill rate shown in Table 1.

[3] For more information, see `http://www.univ-valenciennes.fr/ROI/SID/tdelot/vespa/simulator.html`

Parameters	**Initialization**
Number of physical cells	64
Number of parking lots	8

Number of places for each parking lot	P1 = 50, P2 = 20,
	P3 = 40, P4 = 50,
	P5 = 50, P6 = 20,
	P7 = 50, P8 = 40

Initial load of each parking lot	P1 = 70%, P2 = 85%,
	P3 = 80%, P4 = 66%,
	P5 = 76%, P6 = 65%,
	P7 = 64%, P8 = 75%

Table 2: Parameters considered during the simulations for the parking lots

Once driving on a parking lot, vehicles move at 10km/h whereas their speed is 30km/h elsewhere. Each hour, Q vehicles ($Q = 100$ 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 (what can happen when the number of resources is low), they stop searching and continue exchanging data with the other vehicles until they 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 starts advertising 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 three important criteria when searching for an available parking space:

1. the time needed for each vehicle to find a free space;
2. the percentage of vehicles that actually found a free space within the determined period of time;
3. the percentage of effective information (i.e., percentage of cells indicated by the system to the driver leading to a success in the search of a parking space).

During our experimentations, we considered several elementary strategies, and then combined them into more complex ones:

- *View*: this strategy corresponds to the actual view of a parking space by a driver. 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. In our simulations, a driver of a vehicle is supposed to "see" a free space and park there when the distance between this vehicle and the space is less than 25 meters.
- *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 to the drivers by an infrastructure (i.e., a central server keeping track of all the events occurred). 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 filled by all the events observed in a preliminary simulation of 24 hours. Our goal with this strategy is to evaluate the effect of a "perfect" aggregation data structure containing all occurred events. With this strategy, the driver is guided towards the nearest zone with the highest frequency of parking spaces released.

- *SummaryAggregation*: this strategy also considers the spatio-temporal aggregation data structure but only at the vehicle level. This "embedded" data structure is filled with the events observed by the vehicle since the beginning of the simulation (i.e., located within a predefined radius around the vehicle's location). This range is a parameter of the simulations. More precisely, we chose the values 50% and 25% designating respectively a radius corresponding to half (a quarter of) the radius of the area. With this strategy, the driver is guided towards the nearest zone with the highest frequency of parking spaces released. The *SummaryAggregation* strategy can be used with or without exchanges between vehicles. In the first case, a vehicle does not get any information from the others whereas in the second case, it exchanges (parts of) its data structure with other vehicles located in a range of 200 m of its current location.

- *View+Dissemination*: this strategy combines the *View* and *Dissemination* strategies. A vehicle uses the *Dissemination* strategy first to go towards a potentially free place (provided that no other vehicle reached it before), but will choose any free space observed on its way.

- *View+Dissemination+SummaryAggregation (resp. View+Dissemination+infrastructure)*: this strategy combines *View*, *Dissemination* and *SummaryAggregation* (resp. *Infrastructure*). Thus, when drivers searching for a free parking space do not see any one and do not receive any message from another vehicle releasing its place, the spatio-temporal aggregation data structure (or the infrastructure) is used to select the best area where the parking space should be searched.

For strategies using the aggregation structure (*SummaryAggregation* and *View+SummaryAggregation+Dissemination*), an initialization phase of the aggregation structure precedes the simulations. This corresponds a 24 hours simulation to complete the structure in accordance with the observation range. After this initialization phase, this structure is continuously updated during the simulations. The results presented in the following were obtained by computing the average over 10 simulations for each strategy.

5.3 Qualitative evaluation of the spatio-temporal aggregates

Our first objective with the simulations was to highlight the effectiveness of spatio-temporal aggregates for vehicles searching for an available parking space.

Therefore, Figure 10 shows the results produced by the strategies *View*, *Dissemination* and *Infrastructure* concerning the average time needed by vehicles to find a parking space, the percentage of vehicles that actually found a parking space and the percentage of effective information provided.

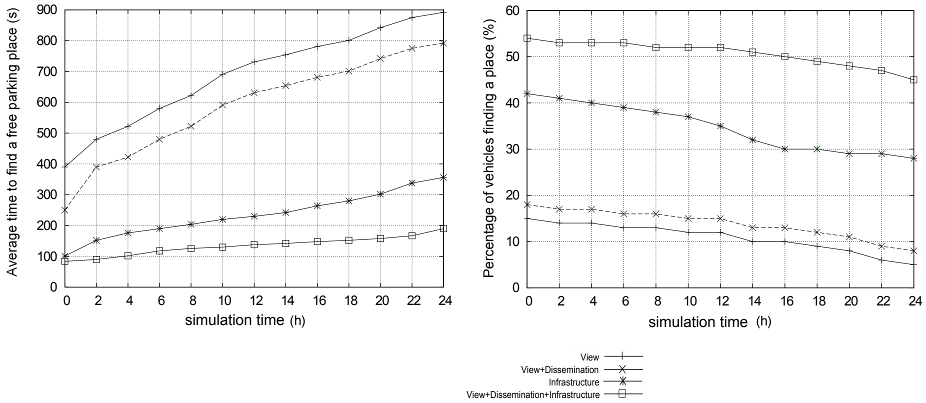


Figure 10: Aggregates vs. dissemination to find an available parking space

In the first part of Figure 10, we observe the upper (*Infrastructure*) and lower bounds (*View*). We first notice 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 parking space and the percentage of effective information decrease over time. The explanation is that the number of vehicles joining 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 2). 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 assistance systems are actually useful and should so be particularly effective in such configurations where it is very difficult for drivers to find an available parking space.

The results presented in Figure 10 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 considering the average time to find a parking space, the percentage of vehicles finding a parking space and the percentage of effective information. Moreover, *View+Dissemination+Infrastucture* is the best strategy showing that the corresponding elementary strategies are complementary.

In Figure 11, we introduce the partial aggregation process at the vehicle level (i.e., the *SummaryAggregation* strategy) and compare it with the strategies already presented in Figure 10. At this stage, we did not consider any exchange of summary between vehicles, the impact of the exchange protocol will indeed be evaluated later in section 5.4. On the contrary, we analyzed the

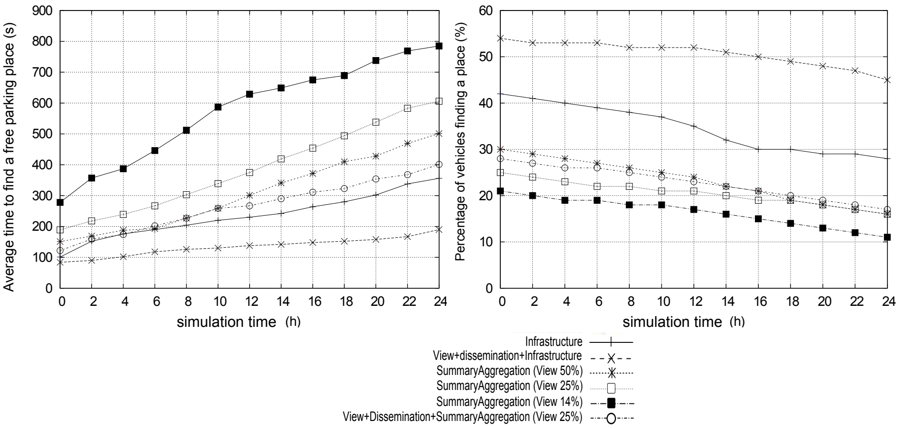


Figure 11: Impact of the observation range

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). By comparing the results of *SummaryAggregation* with a range of 50% with the ones obtained for *Infrastructure*, we notice that even if the "quality" of the data structure is divided by 2 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 complete aggregation process is not mandatory to have actual benefits.

5.4 Evaluation of the exchange protocol

Thanks to the exchange protocol introduced in section 4, (parts of) summaries can be exchanged between vehicles when they encounter each other. In this section, we evaluate the impact of such exchanges on the quality of the aggregates produced (i.e., the effectiveness of the predictions done with this aggregates). Therefore, for each range considered for *SummaryAggregation* (e.g., 50%, 25% and 14%), we compare the results with and without exchanges of aggregates. The exchange of summary between two vehicles occurs when the distance between them is less than 100 meters.

In Figure 12 we introduce the exchanges of summaries between vehicles. Figure 12 shows that performing exchanges between vehicles significantly improves the results obtained with *SummaryAggregation*. By comparing the same strategies with and without exchange, we indeed observe that the use of the exchange protocol increases by 10% the number of vehicles finding an available parking space. 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.

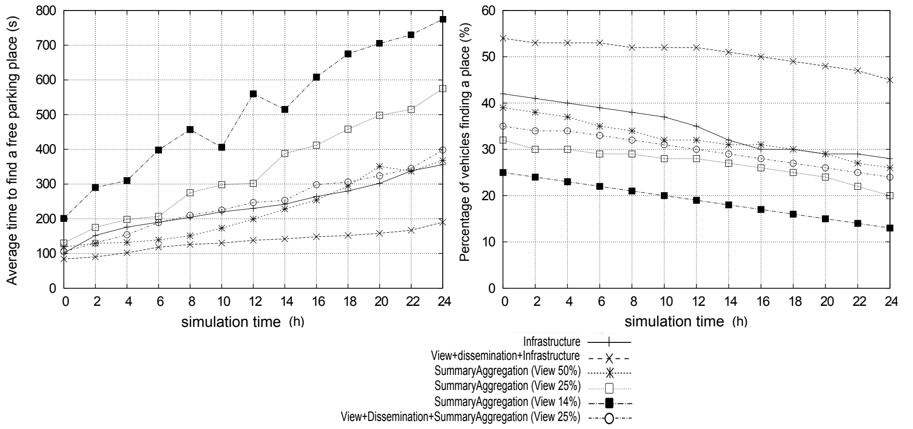


Figure 12: Impact of the exchange protocol

5.5 Study of vehicular exchanges

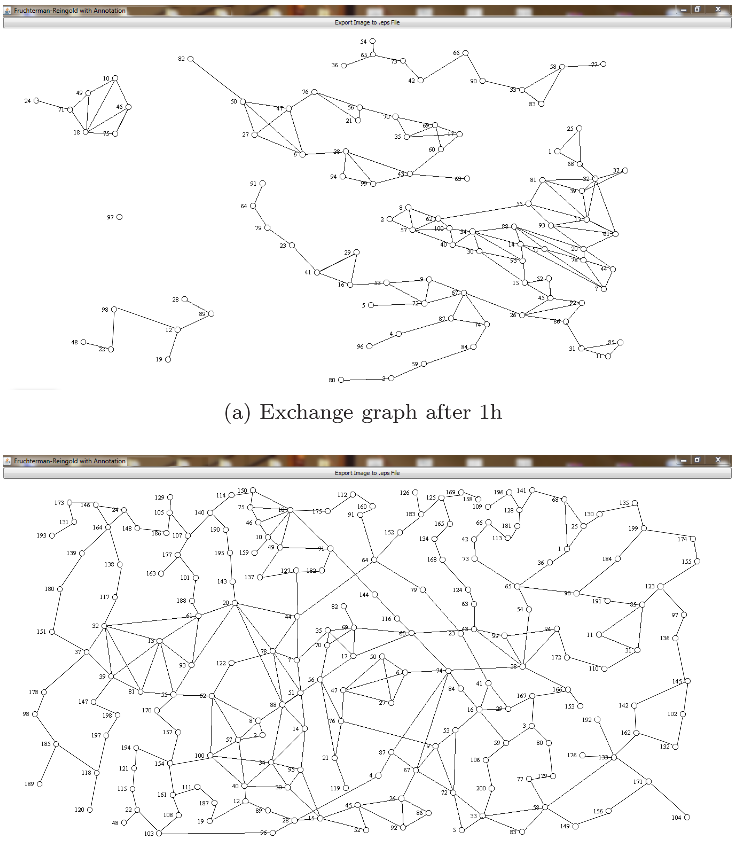
In this section, we present an analysis about the dynamics of exchanges occurring between vehicles. Our goal is to answer three main questions:

1. does the exchange process allow two distant vehicles to actually exchange summaries?
2. how does the exchange process evolve over time?
3. what is the impact of the number of exchanges on the quality of the indications provided to the driver?

To answer these questions we conducted another series of simulations considering the same environment and the parameters defined in Table 2. As for previous simulations, an initialization phase is performed for filling the structure with events observed during 24h. The strategy used to assist the driver is *SummaryAggregation* with a "vision range" of 50%. In the following, we concentrate our study on the variation of three parameters:

1. the duration of the exchange period to observe the evolution of exchanges over time: We therefore considered two periods of time of 1 and 2 hours respectively;
2. the communication range between two vehicles (i.e., the maximum distance between them allowing the exchange of summaries): Again, we selected two communication range of 100 meters and 50 meters respectively.
3. the number of vehicles: every hour, 100 new vehicles enter the simulation environment and start looking for a free parking place during 1000s.

In the following, we represent these exchanges occurred between vehicles during the observation period as an undirected graph. The nodes represent vehicles and the edges the direct exchanges (if any). For the sake of clarity, we represent at most one exchange between two vehicles. In case several exchanges of summaries between the same couple of vehicles occur, only a single edge is represented in the graph. The graphs presented in the following figures are visualized and analyzed using NWT (Network Workbench Tool). NWT takes as input a log file containing the nodes with their identifier and the exchange relations to be visualized and analyzed.


Fruchterman-Reingold with Annotation
Export Image to .eps File
Fruchterman-Reingold with Annotation
Export Image to .eps File

(a) Exchange graph after 1h

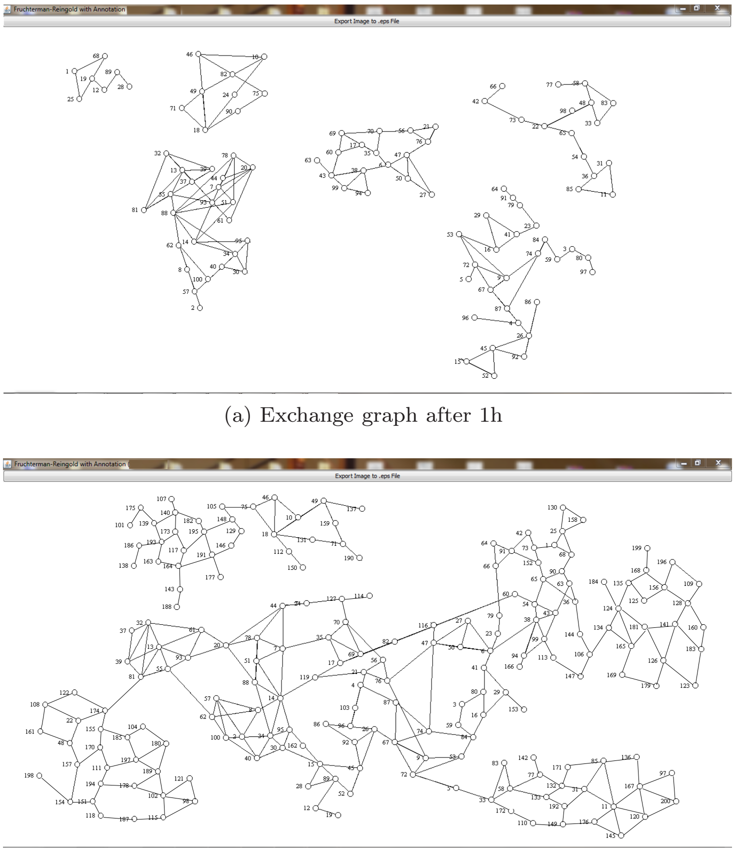
(b) Exchange graph after 2h

Figure 13: Evolution of the exchange graph with a communication range of 100m

Figures 13a and 13b show the interactions between 100 (resp. 200) vehicles in 64 physical cells after respectively one hour and two hours of simulations. The communication range considered in both these figures is equal to 100 meters.

Figures 14a and 14b show the same interactions as in Figure 13 but considering a communication range of 50 meters.

The analysis shows that in both cases (Figure 14 and 13) the number of connectivity classes decreases significantly over time (i.e., from 5 classes to only 1 with a communication range of 100 meters and from 6 classes to 2 with a communication range of 50 meters). Moreover, considering a communication range of 100 meters, the number of edges in the graph increases from 2000 after 1 hour to 7200 edges after 2 hours. Hence, by doubling the exchange period, we observed an enhancement factor of edges over 3,5.


Fruchterman-Reingold with Annotation
Export Image to .eps File
Fruchterman-Reingold with Annotation
Export Image to .eps File

(a) Exchange graph after 1h

(b) Exchange graph after 2h

Figure 14: Evolution of the exchange graph with a communication range of 50m

The parameters observed for the exchanges with a communication range of 100 meters are presented in Figure 13 is given by Table 3. The ones obtained with a communication range of 50 meters are presented in Table 3.

	Nb of nodes	**Simulation time**	**Nb of exchanges**	**min arity**	**max arity**	**avg. arity**	**path length**
Graph 13a	100	1h	2014	0	39	22	8
Graph 13b	200	2h	7259	31	52	41	11

Table 3: Analysis of graphs 13a and 13b

	Nb of nodes	**Simulation time**	**Nb of exchange**	**min arity**	**max arity**	**avg. arity**	**path length**
Graph 14a	100	1h	984	1	24	10	5
Graph 14b	200	2h	4275	20	43	27	9

Table 4: Analysis of graphs 14a and 14b

It is also interesting to observe if the exchanges are limited to neighboring vehicles or if two vehicles initially far away from each other can directly exchange according to their displacements. We therefore examine inter-cell exchanges. Inter-cell exchanges are defined as a direct exchange between two vehicles located in distinct cells at the beginning of the simulation.

	Nb. of vehicles	**Simulation time**	**Distance**	**Nb. of inter-cell exchanges**	**ratio of total exchanges**
Graph 13a	100	1h	100	259	1/7
Graph 14a	100	1h	50	175	1/6
Graph 13b	200	2h	100	1609	1/4
Graph 14b	200	2h	50	837	1/5

Table 5: Inter-cell exchanges

In Table 5 we notice that the number of inter-cell exchanges increases over time. By doubling the period of exchanges, the volume of inter-cell exchanges is multiplied by 7 with a communication range of 100 meters and by 5 considering a communication range of 50 meters. This shows that the aggregated information may be spread everywhere thanks to the exchange protocol. However, a more detailed analysis would be needed to understand in what extent this increase depends on the mobility model of the vehicles.

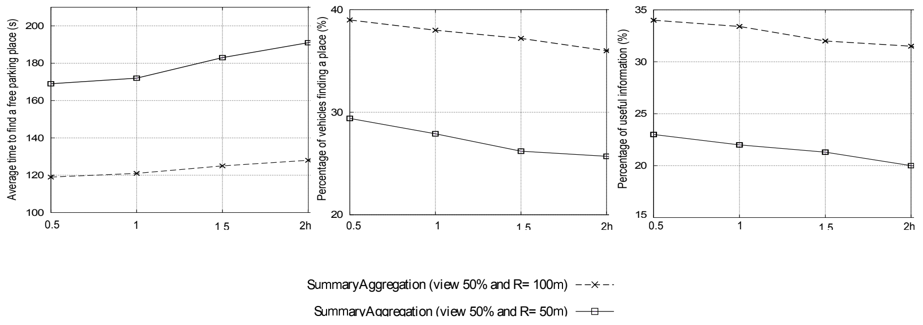


Figure 15: Results with decreasing exchange range

Finally, we study in Figure 15 the influence of the communication range. We therefore observe three parameters: (1) the average time needed to find a free parking space, (2) the percentage of vehicles satisfied (i.e., actually finding an available parking spot) and (3) the percentage of useful information provided by the system (i.e., indications provided by the system and actually helping the driver to park). These results are obtained using the *SummaryAggregation* strategy with a vision range of 50% and a communication range R of 100 meters and 50 meters. The results presented are the average of 10 simulations.

Figure 15 shows that a decrease of the communication range significantly degrades the quality of the assistance provided to the driver using the aggregation structure. The gap is indeed already important after only half an hour of vehicular exchanges.

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. Data aggregation has received a lot of attention in wireless sensor networks [24, 38, 29]. In this context, data aggregation is usually considered as a way to reduce energy consumption [31, 33], which is not a concern in our context.

Data aggregation has also been investigated in vehicular networks, mainly to compress information and reduce bandwidth usage. For instance, the work presented in [35] describes a system, called *TrafficFilter*, in which vehicles collaboratively build a speed profile associated to a 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 [18] present a cluster based aggregation scheme suitable for dissemination of vehicle speeds. Contrary to the previous system, the CASCADE system employs only syntactic, lossless compression of data. At a 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. Differential coding is also used to compress vehicle information in each cluster. Once compressed, the information is then disseminated further. Another approach is presented in [23] where Lochert et al. describe a hierarchical aggregation technique for vehicles' travel times. In this approach, each vehicle broadcasts its travel time between two landmarks along its trip. These travel times are then aggregated hierarchically and broadcasted to provide distant vehicles with an estimate of the travel times along the road segments. Receiving vehicles can thus avoid congested roads (i.e., the roads with larger travel time estimate).

In [32], RLSMP (Region based Location Service Management Protocol) is proposed. It is based on the aggregation of messages according to geographical areas. The goal of the authors is to reduce the number of messages generated for the management of vehicles' location. The authors highlight that aggregation improves scalability, but can also lead to: (1) More packet collisions and so more retransmissions (mainly because of the size of packets exchanged). (2) Longer delays since processing is required before data can be effectively delivered.

Eichler et al. consider in [14] vehicles aggregating data about warnings when they receive multiple messages related to the same event. They also propose the use of invalidation messages when a vehicle did not detect a danger in an area defined as dangerous according to the aggregated information.

Works mentioned previously generally consider data summarization as a method for compressing information and thus save network bandwidth. Data compression and data aggregation are however distinguished in [27] where the authors present *TrafficView*, a system exploiting semantic aggregation. The authors present two techniques for aggregating data: *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 fine grain information may not be required over a long distance. The resulting view of traffic conditions is, thus, customized for each particular vehicle. In the *cost-based aggregation* technique, data is aggregated based on a cost function that depends on the position of the aggregating vehicle.

Different types of aggregation are also studied in [28] where Picconi et al. classify aggregation techniques as either syntactic or semantic. Syntactic aggregation uses a technique to compress or encode the data from multiple vehicles in order to fit the data into a single frame. This results in a lower overhead than sending each message individually. In semantic aggregation, the data from individual vehicles is summarized. For instance, instead of reporting the exact position of five vehicles, only the fact that five vehicles exist is reported. Hence, the message to exchange is much smaller due to a loss of precise data.

Yu et al. [39] present an aggregation technique called Catch-Up that aggregates similar reports generated by vehicles whenever an event occurs (e.g., a change in the traffic conditions). The technique is based on the insertion of a delay before forwarding any report so that similar reports received from surrounding vehicles can be aggregated into a single report.

In [18] authors present *CASCADE*, a technique for accurate aggregation of vehicle data. The goal of CASCADE is to allow a vehicle to obtain an accurate view of upcoming traffic conditions. Vehicles will pass information about traffic conditions ahead of them to vehicles behind them so that these vehicles will have timely notification of upcoming traffic conditions. The local view presents data gathered from primary records, which are sent in signed frames containing a vehicle's position information. The local view is grouped into clusters, which are then used to compact and aggregate the local view data.

In [7], the authors present the protocol LBAG (Location Based Aggregation). In this protocol, data aggregation relies on a hierarchy of static locations instead of considering a tree of nodes that would be particularly difficult to maintain because of the high mobility of vehicles. A Geocast communication protocol is used to transmit a message in a target area.

In [8], the authors describe a framework to efficiently summarize several streams joined by a relationship between one another. Summaries are built, which give information both on each stream individually, as well as on their relationship for any given time horizon. To realize this summary, three techniques where used in the summary structure: the first one is the micro cluster [1] that makes use of the Cluster Feature Vector (CFV) aggregate [30]. The second one consists in dividing treatment between an online part producing snapshots of the system state, and an offline part analyzing these snapshots [1]. Finally the third technique relies on the use of Bloom Filters [3].

7 Conclusions and Future Work

In this article, we presented an aggregation structure for events produced and exchanged in vehicular networks. This structure is based on a two-level spatio-temporal model that allows to manipulate the same physical repository for all vehicles. The important properties of our data structure reside in its capacity to be exchanged without loss of information and to be duplicate insensitive. Moreover, the storage space required for our aggregation structure is limited provided that the number of temporal dimensions remains controlled. Besides, the complexity to access the structure is also efficient (logarithmic or linear).

Through numerous simulations, we have proved the effectiveness of our solution under different assumptions. 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 anticipate and optimize the connection time between two vehicles willing to exchange (parts of) their summaries.

Acknowledgments

The present research work has been supported by Institut Telecom with a Future and Rupture grant, the International Campus on Safety and Intermodality in Transportation (CISIT), the European Community, the Regional Delegation for Research and Technology, the Ministry of Higher Education and Research and the National Center for Scientific Research. The authors gratefully acknowledge the support of these institutions.

References

[1] C. Aggarwal, J. Han, J. Wang, and P. Yu. A framework for clustering evolving data streams. In 29th Conference on Very Large DataBases (VLDB), 2003.

[2] D. Ayala, O. Wolfson, B. Xu, B. DasGupta, and J. Lin. Parking in competitive settings: A gravitational approach. In 13th International Conference on Mobile Data Management (MDM), pages 27–32, 2012.

[3] B.H. Bloom. Space/time trade-offs in hash coding with allowable errors. JSDA Electronic Journal of Symbolic Data Analysis, 13(7):422 – 426, 1970.

[4] P. Caballero-Gil, J. Molina-Gil, and C. Caballero-Gil. Data aggregation based on fuzzy logic for vanets. In Computational Intelligence in Security for Information Systems, Lecture Notes in Computer Science, pages 33–40. Springer, 2011.

[5] N. Cenerario, T. Delot, and S. Ilarri. Dissemination of information in inter-vehicle ad hoc networks. In Intelligent Vehicles Symposium (IV), pages 763–768. IEEE Computer Society, June 2008.

[6] N. Cenerario, T. Delot, and S. Ilarri. A content-based dissemination protocol for VANETs: Exploiting the encounter probability. IEEE Transactions on Intelligent Transportation Systems, 12(3):771–782, September 2011.

[7] C. Chen. Location-based data aggregation in mobile ad hoc networks. Master's thesis, Institute fur Parallele und Verteilte Systeme, Stuttgart, 2003.

[8] B. Csernel, F. Clerot, and G. Hébrail. Summarizing a 3 way relational data stream. In Workshop on Data Stream Analysis, March 2007.

[9] B. Defude, T. Delot, S. Ilarri, J.-L. Zechinelli, and N. Cenerario. Data aggregation in VANETs: the VESPA approach. In Workshop on Computational Transportation Science (IWCTS), July 2008.

[10] T. Delot, N. Cenerario, and S. Ilarri. Vehicular Event Sharing with a mobile Peer-to-peer Architecture. Transportation Research - Part C (Emerging Technologies),18(4), 18(4):584–598, August 2010.

[11] T. Delot, S. Ilarri, N. Cenerario, and T. Hien. Event sharing in vehicular networks using geographic vectors and maps. Mobile Information Systems, 7(1):21–44, 2011.

[12] T. Delot, S. Ilarri, S. Lecomte, and N. Cenerario. Sharing with caution: Managing parking spaces in vehicular networks. Mobile Information Systems, 9(1):69–98, 2013.

[13] T. Delot, S. Ilarri, N. Mitton, and T. Hien. GeoVanet: A Routing Protocol for Query Processing in Vehicular Networks. Mobile Information Systems, 7(4):329–359, 2011.

[14] S. Eichler, C. Merkle, and M. Strassberger. Data aggregation system for distributing inter-vehicle warning messages. In Conf. on Local Computer Networks, 2006.

[15] N.E. Faouzi, H. Leung, and A. Kurian. Data fusion in intelligent transportation systems: Progress and challenges - a survey. Information Fusion, 12(1):4–10, 2011.

[16] P. Flajolet and G. N. Martin. Probabilistic counting algorithms for data base applications. Journal of Computer and System Sciences, 31(2):182–209, 1985.

[17] R. Glabb, L. Imbert, G. Jullien, A. Tisserand, and N. Veyrat-Charvillon. Multi-mode operator for sha-2 hash functions. Journal of Systems Architecture, 53(2-3):127–138, February 2007.

[18] K. Ibrahim and M. C. Weigle. CASCADE: Cluster-based accurate syntactic compression of aggregated data in VANETs. In Workshop on Automotive Networking and Applications (AutoNet), pages 1–10, December 2008.

[19] G. Karagiannis, O. Altintas, E. Ekici, G. J. Heijenk, B. Jarupan, K. Lin, and T. Weil. Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. IEEE Communications Surveys and Tutorials, 13(4):584–616, 2011.

[20] G.J.M. Kruijff, J.D. Kelleher, and N. Hawes. Information fusion for visual reference resolution in dynamic situated dialogue. In E. André, L. Dybkjaer, W. Minker, H. Neumann, and M. Weber, editors, Perception and Interactive Technologies (PIT). Springer Verlag, 2006.

[21] J. Kulik, W. Heinzelman, and H. Balakrishnan. Negotiation-based protocols for disseminating information in wireless sensor networks. Wireless Networks, 8(2/3):169–185, 2002.

[22] C. Lochert, B. Scheuermann, and M. Mauve. Probabilistic aggregation for data dissemination in vanets. In Workshop on Vehicular Ad Hoc Networks (VANET), pages 1–7, 2007.

[23] C. Lochert, B. Scheuermann, C. Wewetzer, A. Luebke, and M. Mauve. Data aggregation and roadside unit placement for a vanet traffic information system. In Workshop on VehiculAr Inter-NETworking (VANET), pages 58–65, September 2008.

[24] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. Tag: a tiny aggregation service for ad-hoc sensor networks. ACM SIGOPS Operating Systems Review, 36(SI):131–146, October 2002.

[25] S.S. Manvi, M.S. Kakkasageri, and J. Pitt. Multiagent based information dissemination in vehicular ad hoc networks. Mobile Information Systems, 5(4), page 363–389, 2009.

[26] H. Mousannif, I. Khalil, and S. Olariu. Cooperation as a service in vanet: Implementation and simulation results. Mobile Information Systems, 8(2):153–172, 2012.

[27] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode. TrafficView: Traffic data dissemination using car-to-car communication. ACM SIGMOBILE Mobile Computing and Communications Review, 8(3):6–19, July 2004.

[28] F. Picconi, N. Ravi, M. Gruteser, and L. Iftode. Probabilistic validation of aggregated data in vehicular ad hoc networks. In Workshop on Vehicular Ad Hoc Networks (VANET), pages 76–85, 2006.

[29] B. Przydatek, D. Song, and A. Perrig. Sia: Secure information aggregation in sensor networks. In Conference on Embedded Networked Sensor Systems (SenSys), pages 255–265, November 2003.

[30] T. Zhang R. Ramakrishnan and M.Livny. Birch: an efficient data clustering method for very large databases. In Conference on Management of Data (SIGMOD), 1996.

[31] R. Rajagopalan and P. Varshney. Data aggregation techniques in sensor networks: a survey. IEEE Communications Surveys & Tutorials, 8(4):48–63, 2006.

[32] H. Saleet and O. Basir. Location based message aggregation in vehicular ad hoc networks. In IEEE Global Communications Workshops, pages 1–7, November 2007.

[33] Dietzel Stefan, Schoch Elmar, Bako Boto, and Kargl Frank. A structure-free aggregation framework for vehicular ad hoc networks. In Workshop on Intelligent Transportation (WIT), March 2009.

[34] Y. Tao, G. Kollios, J. Considine, F. Li, and D. Papadias. Spatio-temporal aggregation using sketches. In 20th Conference on Data Engineering (ICDE), pages 214–225, 2004.

[35] M. van Eenennaam and G. Heijenk. Providing over-the-horizon awareness to driver support systems. In Workshop on Vehicle-to-Vehicle Communications, 2008.

[36] Vasilis Verroios, Vasilis Efstathiou, and Alex Delis. Reaching available public parking spaces in urban environments using ad-hoc networking. In 12th International Conference on Mobile Data Management (MDM), pages 141–151. IEEE Computer Society, 2011.

[37] B. Xu, A. M. Ouksel, and O. Wolfson. Opportunistic resource exchange in inter-vehicle ad-hoc networks. In Conference on Mobile Data Management (MDM), pages 4–12, January 2004.

[38] Yang Yi, Wang Xinran, Zhu Sencun, and Cao Guohong. Sdap: A secure hop-by-hop data aggregation protocol for sensor networks. ACM Transactions on Information and Systems Security, 11(4):18:1–18:43, July 2008.

[39] B. Yu, J. Gong, and C.-Z. Xu. Catch-up: A data aggregation scheme for vanets. In Workshop on VehiculAr Inter-NETworking (VANET), pages 49–57, September 2008.