Daniel Burgstahler, Tobias Meuser, Ulrich Lampe, Doreen Böhnstedt and Ralf Steinmetz: ProbSense.KOM: A Probabilistic Sensing Approach for Gathering Vehicular Sensed Data. In: Proceedings of the 5th IEEE International Conference on Mobile Services (MS2016), p. 1-8, June 2016

ProbSense.KOM: A Probabilistic Sensing Approach for Gathering Vehicular Sensed Data

Daniel Burgstahler, Tobias Meuser, Ulrich Lampe, Doreen Böhnstedt and Ralf Steinmetz
Multimedia Communications Lab (KOM), Technische Universität Darmstadt, Darmstadt, Germany
Email: {firstName.lastName}@KOM.tu-darmstadt.de

***Abstract*—Advanced driver assistance systems (ADAS) improve safety, energy efficiency and driver comfort. Such systems are commonly based on sensor data; however, sensor range is physically limited. A way to extend the sensing range is to share sensor reading with others, i. e., vehicles and infrastructure services. Since direct vehicle communication is not widely deployed and vehicles are often not driving in direct communication range, communication has to be realized via cellular networks. Due to high costs for cellular communication, the transmission of sensed data has to be efficient and the amount of transmitted data must be minimized. As possible solution, we introduce a concept of probabilistic data transmission for vehicular sensed data. The system divides the map into geographic cells, and a probabilistic model is managed for each geographic cell individually. We are able to achieve a reduction in data transmission volume of up to 50 % in comparison to opportunistic approaches.**

***Index Terms*—probabilistic sensing, vehicular data collection**

I. Introduction

Advanced Driver Assistance Systems (ADAS) support to increase driving comfort, economy and safety. These systems are based on local sensor readings, providing information about vehicle status and vehicle surroundings. An important factor is the detection coverage, since a larger sensing range enables more predictive systems. However, due to the physical limitation, the sensor detection range is limited. A possible solution could be to share local sensed information because it is also potentially valuable for other vehicles. This could either be done by infrastructure based or direct vehicle to vehicle (V2V) communication.

Direct V2V communication might be the first choice to share information in the direct environment, especially for safety critical information with low latency requirements. To provide a central, holistic, and up-to-date information base and also to be able to cope with low traffic density, a centralized information sink is desirable. Here, cellular communication is the appropriate technology and an according system can benefit from an already existing high network coverage. However, as the number of potentially connected vehicles is very large, an intelligent data collection management is necessary to minimize data traffic. This is necessary for both preventing network overload and minimizing transmission costs.

Classic approaches consider optimizations for complete data transmission like local pre-processing or clustering with aggregation to reduce data traffic. Another strategy is an incomplete transmission model. Due to the relatively high number of potential mobile sensors, i. e., the connected vehicles, data collection will have a high redundancy. The aim is to reduce the data traffic and still satisfy certain quality of service (QoS) parameters, e. g., maximum detection latency. A classic use case is to detect changes in the road network, including changes in traffic signs. Changes can be detected several times within a relatively short period of time.

By introducing a transmission probability, the amount of transmissions can be reduced. Therefore, we suggest a probabilistic data collection strategy within this scenario. Since the vehicular traffic volume is spatio-temporally changing, transmission probabilities also have to be adapted. Our approach considers transmission probabilities separately for each property and conducts the management based on geographic cells. The developed model uses the data quality indicators detection latency, i. e., the time an event takes to be stored in the database, and data density, i. e., the amount of events per hour and square kilometer in order to reduce the data traffic. To evaluate this approach, we used the SUMO traffic simulator with an extended scenario of the *TAPAS Cologne*[1] data set [1]. We were able to show a data traffic reduction of about 50 % with a defined detection latency of 10 minutes. Since this improvement is dependent on the desired data quality, the result can be further improved if a lower data quality level is acceptable.

The remainder of this paper is structured as follows: First, we summarize work related to mobile sensing and sensing data sharing in Section II, followed by a system overview and concept description in Section III. In the following, we provide a formal specification of our approach in Section IV and give a detailed description of the server side and client side in Sections V and VI. We present and discuss our evaluation results in Section VII and finally conclude our work with a summary and outlook in Section VIII.

II. Related Work

First of all, the considered scenario can be classified into the general field of mobile sensing. Within this field, smartphones are typically considered as sensing devices because of their versatility and large amount of sensors [2]. Another related field is Mobile Crowd Sourcing (MCS), which utilizes a huge amount of mobile devices to build large scale sensing applications [3]. To distinguish between different sensing applications, we categorize sensing models using three criteria: The amount of subjects participating, the degree of human participation,

[1] http://sumo.dlr.de/wiki/Data/Scenarios/TAPASCologne

and the treatment of collected sensor data. The amount of subjects participating enables to divide the sensing into three subcategories, namely personal, group, and community sensing. Within the considered scenario primarily community sensing, that is investigated by MCS, is of interest. Concerning the required user interaction, one can distinguish between participatory sensing and opportunistic sensing. Participatory sensing is a sensing model that applies user interaction to collect data and focuses on tools to help users to share, search, interpret, and verify information [2], [4]. In the considered scenario a participatory sensing model is not applicable due to active involvement of the user. If no active participation is required, the model is named opportunistic sensing. This is a big advantage in our scenario and commonly used in traffic monitoring applications, i. e., vehicles transmit data about their environment themselves [5]. According to Shin et al. , this sensing model has to cope three basic challenges [5]: To ensure data quality and integrity, to protect users privacy, and to consider an efficient data transmission. Depending on the treatment of information collection, a model can further be categorized into probabilistic and deterministic sensing. In the latter one, all sensed data are sent. Thus, complex maintenance or processing can be neglected, but network traffic might become a problem.

In probabilistic sensing, sensed information will only be transmitted with a certain probability, which might depend on several factors, e. g., the distance between the sensor and the measured event (shadow fading) [6]. Considering the sensing device, vehicles are in focus of interest within our context. Vehicles are equipped with a variety of sensors that produce information that is potentially of interest for other vehicles in the direct surroundings. This information sharing enables to extend the size of the perceived environment. For this information exchange, vehicle to vehicle (V2V) communication has been introduced [7]. Classical use cases for V2V communication are safety systems like, e. g., a collision warning system [8]. To extend the transmission range, several approaches exists that also extend the sensing model used to community sensing [2]. However, in general, V2V communication is limited in transmission range and is poorly applicable in sparse traffic situations.

An important aspect in information distribution is clustering of sensing nodes, i. e., grouping nodes in geographical vicinity according to rules [9]. The basic concept is to reduce the overall transmission costs by collecting all data at a so called cluster head and transmitting it conjointly to the sink, e. g., a cloud service. Clustering algorithms basically focus on the selection of the cluster head, which might also summarize and compress the data before transmission.

A basic algorithm is named *lowest ID (LID)*. Each node is assigned a unique *ID*. Nodes broadcast their *ID* and allocate themselves to the node with the lowest *ID*. The node with the lowest *ID* is selected to be the cluster head [10]. Within *highest degree clustering (HD)*, the number of clusters is minimized by a cluster head selection based on the nodes with the highest number (degree) of nodes in direct communication range [11]. Another clustering approach is the *weighted clustering algorithm (WCA)* that is based on a performance indicator of several properties like node degree, transmission power, mobility, and battery power [12].

Lowest Relative Mobility Clustering Algorithm (MOBIC) is an approach to use *LID* in vehicular networks efficiently [13]. MOBIC replaces the id with a performance indicator (relative speed to neighbors) and takes the node mobility (compared to neighbors) into account. The *MOBIC* concept has been the basis for several further developments. A different approach is introduced in the CONVERGE project that uses a beacon-based clustering for performance improvement [14]. It is probability based to balance network traffic. Each vehicle starts with a probability $p = 1$ to become the cluster head. In an adaption phase, each node communicates with its neighbors if it decides to become the head of the cluster. Each time a vehicle receives such a beacon, the own probability to become the cluster head is lowered.

Another approach to reduce data traffic is local data selection by discarding unnecessary data and simultaneously ensure to only harm data quality within a certain threshold. The send-on-delta reporting strategy only considers sensor values with a certain measurement difference [15]. This method is mainly considered in stationary wireless networks and the performance depends on the size of the delta and the change rate of the measurements. As improvement, Suh developed an approach that relies on the prediction of sensor values [16]. A linear prediction method is used to estimate the sensor values and these are only transmitted if the difference between the predicted and real sensor value exceeds a certain delta.

Chu et al. developed the *Ken* strategy that is based on a central sink node [17]. The sink answers to external queries by using a predicted value of a replicated dynamic probabilistic model. All sensor nodes are maintaining the same probabilistic model as the sink and transmit updates to the sink if values differ a certain threshold. The approach guarantees value accuracy within a certain range and the prediction outperforms linear progression of the send-on-delta mechanism.

Deshpande et al. introduced an approach for collecting correlating values using a probabilistic model [18]. Similar to *Ken*, a central server collects required information and answers external queries, but filters false sensor value transmission. In comparison to *Ken*, the sensor nodes do not transmit data independently, but are requested by the server if the uncertainty for a specific value is high. This releases the sensor nodes from predicting values themselves as well as receiving the prediction function.

Hull et al. developed *CarTel*, where nodes are not allowed to transmit data independently, but only if they are requested by the server – named portal – itself [19]. They use opportunistic wireless connectivity, e. g., Wi-Fi, to communicate either with the portal or with other available devices delivering the data. Nodes do not transmit a continuous data stream; data is only transmitted once they were externally requested. Mobile applications can query data from the portal, then the mobile

nodes are requested to stream the required data to the portal server.

The CafNet data delivery mechanism [20] enhances data on vehicle side instead of transmitting raw sensor data. At the portal data is stored in a relational database. Thus, the request latency for already stored information is quite low.

In general, an approach to reduce the amount of data to be transmitted to the server is to do a local pre-processing that exercises the possibility to reduce the data traffic by compressing. As example, Li et al. utilize the data sparsity of the collected information [21]. This is done using an nonlinear algorithm to reconstruct the compressed data and an algorithm to perform random sampling on a sparse basis. The CS framework they propose can be utilized to compress the information sent in the Internet of Things (IoT) context.

In summary, the range of sensor data collection approaches is very large. Most data collection approaches in the vehicular environment are using clustering to optimize data transmission to a central server. This assumes the considered vehicles to be equipped with vehicular ad-hoc communication technology – which is not the case today. However, vehicles are increasingly getting connected via cellular networks. Approaches that discard sensed data mostly focus on continuous sensed data or the predictability of prospective sensed values. Other available strategies for incomplete data transmission are based on static network topologies and thus not suitable for the mobile scenario. To the best of our knowledge, our work is the first that considers a dynamic probabilistic model based on geographic cells to reduce the overall data traffic in a central vehicular data collection scenario.

III. System Overview & Concept

Our aim within this work is a system that provides the means for a central up-to-date map database based on cellular communication technology. Communication has to be minimized, since it is potentially costly. A general system overview is depicted in Figure 1. Vehicles serve as mobile sensors and are connected via a cellular link to the server side. For realization of a scalable connection management, we make use of a MQTT based publish subscribe system for the interconnection of the mobile clients and the server side. This brings the further advantage of asynchronous communication that decouples bidirectional information flow between client and server side which can be beneficial in case of varying connectivity. A basic assumption in our model is that a certain detection latency can be tolerated at the server side. Thus, we consider an incomplete transmission model, that transmits the collected data with a certain probability. The transmission probability is centrally managed, not only for each property individually, but also separately for each geographic region, organized as geo cells. This allows to adapt to different traffic densities. Within each geo cell, an according probability matrix is distributed. This describes which sensor data to be considered and the according probability to transmit.

In general, information might be sensed very often in dense traffic situations. To reduce this redundancy the transmission is

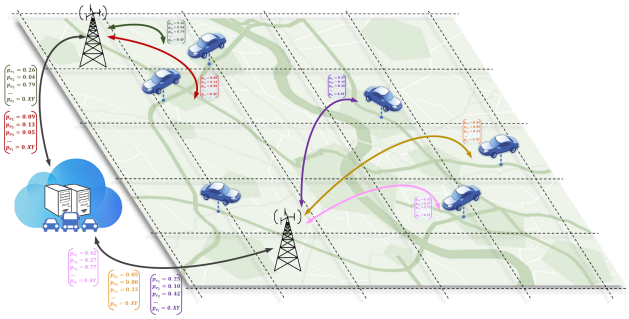

Figure 1: System overview.

controlled by a certain probability. Based on the traffic density a probability for information transmission can be calculated at the server side to ensure a maximum tolerable detection latency. This detection latency describes the time from information occurrence, e. g., a speed limit change, to the registration in the map database, including a predefined redundancy for reliability.

We differentiate between singular events, e. g., a traffic sign change, and continuous events, e. g., temperature sensing. By lowering the probability of a sensed event to be sent, it is possible to control the time till the event is stored at the server. This time shall approach a self defined latency, which is assumed to be sufficient for the certain event type. Thus, the probability is based on the quotient of tolerated detection latency and necessary redundancy. In case of continuous events data density, which defines the amount of events transmitted per square kilometer and hour, is used as parameter to control the transmission probability.

IV. Formal Specification

The whole map is divided into virtual geo cells. Each geo cell $m \in M$ can be considered independent from other cells and thus optimization is processed independently. It is processed for every adjustment period t independently, while T denotes the set of all periods since system start. The set of all event types is E. The number of changes of the probability for an event e is defined as n_e and τ as the required redundancy. The size of an information message, containing the event information, is given with x_e, while the size of a control message is b_e. The probability $p_{e,t} \neq 0$ determines if an information e is sent in the considered geo cell. The number of events e recognized by the vehicles is $c_{e,t}$ and is not influenced by the probability. The total amount of data traffic produced for event e in period t in a geo cell is $F_{e,t}$. Singular events are a subset $E_S \subseteq E$.

For each singular event a maximum latency $l_{max}(e)$ is set, after which the event should be stored in the database in $\rho\,\%$ of all cases. L_t defines the latency of all events measured in the adjustment period t. The $\rho\,\%$ quantile is denoted by $l_{t,\rho}$. Continuous events are also a subset $E_C \subseteq E$. For continuous events the density δ_e is given. The density measured in the period t for an event e is defined as $\delta_{e,t}$.

In order to prevent the infinite sum over all timeslots to go up to infinity, an exponential function may not be considered.

The chosen solution for singular events is based on a quadratic punishment, which converges definitely. But compared to continuous events, in which a lower density is viable for a short amount, an exceeding of the maximum latency requires additional punishment. Therefore, the balancing function starts hops over 100 units in the top once the 99% quantile exceeds the maximum latency.

$$\forall t \in T, e \in E : \min\left(F_{e,t} \times \epsilon_e\right) \quad \text{with} \tag{1}$$

$$\forall e \in E_S : \epsilon_e = \begin{cases} \frac{100}{l_{max}(e)^2} \times l_\rho & \text{if} \quad t, \rho > l_{max}(e) \\ 1 & \text{else} \end{cases}$$

$$\forall e \in E_C : \epsilon_t = \begin{cases} 2 \times \left(\frac{100}{\delta_{av}}\right)^2 \times \left(\delta_t - \delta_{av}\right)^2 + 1 & \text{if} \quad \delta_t > \delta_{av} \\ 1 & \text{else} \end{cases}$$

$$\forall e \in E : F_{e,t} = \left(c_{e,t} \times p_{e,t} \times x_e + n_e \times b_e \times va_t\right)$$

In a certain geo cell m, there might be both a street with very heavy and very light traffic. Surely, the balancing function will be at very high values due to the exponential part of the function. Anyway, then the balancing function will stay at very high values and needs to be optimized around these high values.

V. Server Side

The server side is responsible for balancing the amount of data sent by the vehicles and storing the received information in a database. This is done by setting the probabilistic decision model for each geo cell. Thus, the amount of transmitted data is managed by the according probabilities, which also effect the latency of the received data.

Balancing: Statistically, the amount of data can be cut in half by setting the sending probability to half. The opposite holds when increasing the sending probability with a maximum of $p_{e,t} = 1$. If the resulting data rate of $p_{e,t} = 1$ is still not high enough, then the vehicle density is too low. The main task of the server side is to balance the probabilities, such that the latency of singular events and the density of continuous events are fulfilled. For continuous events, the received density δ_e can be easily measured. However, latency $l(e)$ of singular events cannot be directly be measured. Here, we use the redundancy to approximate the latency. If $a(s \in e, t)$ is the amount of measurements of the same event s in a period t, and I is the set of transmission times for the event, the latency can be approximated. It is the average time between the the first and the last event transmission, multiplied by the required redundancy τ. Assuming relatively homogenous traffic, the latency l is approximated using Equation 2.

$$l = \frac{I_{a(s \in e,t)} - I_1}{a(s \in e, t) - 1} \times \tau \tag{2}$$

This approximation is more exact the higher a_i is. However, the average latency measured by the server is not necessarily the theoretic average latency for the specific probability. This is unproblematic if the average latency measured is higher than the theoretical average latency; the probability would be chosen too high, and the requirements would still be fulfilled. If the measured latency is lower, however, the latencies for an upcoming event cannot be assured to be lower than the maximum latency. To address this issue, we set a limit of how many events need to be transmitted and measured in a geo cell before the probability is adjusted. Furthermore, we divide each geo cell into four sub cells, determine the latency separately, and only use the highest value. A problem in using a probabilistic model is variation in results, which should be mitigated. Several models are available to decrease the variation, most of them being based on past values. Examples are averaging, the moving average, or the method of least squares that is used in our implementation. Because changes on the probability only affect the latency and traffic in future periods, a way to predict the latency and traffic of the next region needs to be considered. A bottom-Up approach is used, since the predictions are calculated for each of the sub cells separately and joined in the geo cell afterwards. The method of least squares is ascribed to Gauss and is – among others – described by Stigler [22]. The idea is to create a prediction graph such that the sum of the squared distance between the measured value and the function value is minimal (cf. Equation 3, in which S needs to be minimized).

$$S = \sum_{i=0}^{n} r_i^2 \quad \text{with} \quad r_i = y_i - f(x_i) \tag{3}$$

Dependent on the scenario, different functions can to be used for $f(x)$. Choosing a linear function for $f(x)$ leads to Equation 4 [23]. To use this formula, the amount of considered values needs to be chosen accordingly to detect the trends.

$$x^*_{T+1} = a_0 + a_1 \times (t+1) \quad \text{with} \tag{4}$$

$$a_1 = \frac{\sum_{t=0}^{T}\left(t - \overline{t}\right)\left(y_t - \overline{y}\right)}{\sum_{t=0}^{T}\left(t - \overline{t}\right)^2} \quad \text{and} \quad a_0 = \overline{y} - a_1 \times \overline{x}$$

Where T is the current period of time, g the values used to calculate the average, x_t the value of period t and x^*_t the predicted value for period t. According to Krucker, an exponential function can be used in the least squares method easily by transforming the exponential function to a linear function [23]. If a standard exponential function like $y = b \times e^{ax}$ is used, this is done the following way:

$$Y = \ln y = \ln\left(b \times e^{ax}\right) = \ln b + ax = AX + B \tag{5}$$

$$\text{with} \quad B = \ln b \quad \text{and} \quad A = a$$

After calculating A and B for the linear function, those can be converted to the exponential function using $a = A$ and $b = e^B$. In order to use the method of least squares, the unchanged values for the latency and the data traffic need to be used. To approximate these raw values, the measured data traffic can be divided by the respective probability to get the raw data traffic and the measured latency can be multiplied with the respective probability to get the raw latency. Our preliminary

results have shown that this method only works if the amount of connected vehicles is increasing not too fast. This is due to the probability propagation strategy, described below, which leads to wrongly calculated estimations. A small change needs to be applied to use the exponential function conveniently. As there is no y-offset defined in Equation 5, the values used to calculate this function $f(0) = 1$ should hold as for every exponential function. Therefore, the first value is set to one and the difference between the actual first value and one is subtracted from the remaining values.

Singular Events: We have chosen to use the average latency to perform probability adjustments, since a direct adjustment has led to unfavorable results due to the high variance in the received latency values. Assuming that $99\,\%$ of the data shall be transmitted within the given latency, it is possible to calculate the average latency and the latency which is undercut by $99\,\%$ of the incoming events. However, the chosen measurement period needs to be long enough to ensure that most of the events have arrived at the server at least two times. In the following, the time between two vehicles passing a specific event type on average is named *passing time*.

The process on the server side works as follows: After the measurement period, the server calculates the average latency of all incoming events. Then, the theoretic average for the latency is calculated to determine the *passing time*. Equation 6 is used to determine the ratio between theoretic average latency and the passing time. Each summand consists of the probability that the event has been transmitted at a latency l and the latency l itself.

$$\bar{l} = \lim_{n\to\infty} \sum_{l=\tau}^{n} \left[\binom{l-1}{\tau-1} \times p_{e,t}^{\tau} \times (1-p_{e,t})^{l-\tau} \times l \right] \quad (6)$$

The measured average latency is now divided by this ratio to determine the *passing time*. Using the adjusted *passing time*, the server determines a probability at which $99\,\%$ of all events are below the maximum latency. The latency behavior dependent on the transmission probability is given in Table I. Figure 2 shows the behavior of the probability dependent on the latency. In this example, one unit is the time between two vehicles detecting the event. As expected, the probability to receive the event after a short delay is much higher for high transmission probabilities. Moreover, Figure 2 shows the cumulative probability for a certain latency. Of course, the probability curve hits the $99\,\%$ threshold sooner the higher the transmission probability is.

Table I: Latency behavior at different probabilities.

Transmission Percentage	10 %	25 %	50 %	75 %	90 %
Ratio between average latency and passing time	50	20	10	6.7	5.6
Ratio between maximum latency (99% below) and passing time	113	43	19	11	8

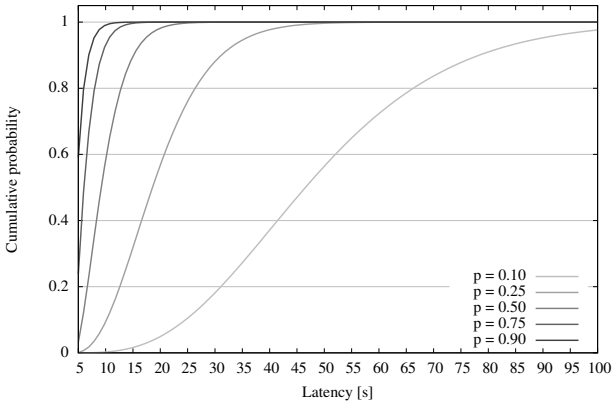

Figure 2: Overview of probabilities depending on latency.

Convergence proof: It is not obvious that Equation 6 converges, so we provide a proof. First we put all constant parts out of the sum. With that, Equation 6 changes to:

$$\bar{l} = p_{e,t}^{\tau} \times (1-p_{e,t})^{-\tau} \times \lim_{n\to\infty} \sum_{l=\tau}^{n} \left[\binom{l-1}{\tau-1} \times (1-p_{e,t})^{l} \times l \right]$$

We choose a C with $C \times (1-p_{e,t}) < 1$ and $C > 1$. C does exist because $(1-p_{e,t}) < 1$. It exists an l_0 with:

$$\binom{l-1}{\tau-1} \times l < C^l \quad \forall\ l \geq l_0 \quad (7)$$

This is true due to the fact that l_1 exists for

$$\left(\frac{(l-1)\times e}{\tau-1}\right)^{\tau-1} \times l \leq l^{\tau} \times \left(\frac{e}{\tau-1}\right)^{\tau-1} < C^l \quad \forall\ l \geq l_1$$

$$\text{and} \quad \binom{n}{k} \leq \left(\frac{n\times e}{k}\right)^{k}$$

Therefore we divide the sum into two parts, one sum from τ to l_0 and one part from $l_0 + 1$ to ∞. Obviously, the first sum has a fixed value. With Equation 7 the remaining part of the sum can be displayed the following:

$$\lim_{n\to\infty} \sum_{l_0}^{n} \left[\binom{l-1}{\tau-1} \times (1-p_{e,t})^{l} \times l \right] <$$

$$< \lim_{n\to\infty} \sum_{l_0}^{n} \left[(1-p_{e,t})^{l} \times C^{l} \right]$$

Per definition $(1-e_e) \times C < 1$ holds, the given equation is a geometric series with a basis smaller than 1, which converges.

Continuous Events: Continuous events do not have a defined appearance date. Therefore, there is no way to calculate the probability based on the latency. Instead, the probability is solely dependent on the amount of incoming events. Because some norm to measure the amount of events required for a region needs to be found, a new variable is introduced, the data density, which equals the amount of events divided by the region size and time. The unit used for the density is

the amount of events per square kilometer and hour. Here, the transmission probability is calculated by comparing the requested density to the actual density and executing a linear transformation.

Realization: Due to the costs of the probability adjustment itself, a probability update should not always be propagated. An adjustment is only performed if the expected data traffic savings are higher than the expected produced traffic for the probability update. The calculation considers the data traffic within one measurement period and the change rate, which is an indicator for the frequency of probability adjustments. In contrast, a probability adjustment is always performed if the newly calculated probability is higher than the current transmission probability, because not performing that update would harm data quality. Algorithm 1 shows the decision

Algorithm 1 Probability adjustment algorithm

```
newProb ←calcProb(actual, expected);
if newProb > oldProb then adjustProb();
else
    if newProb < oldProb then
        savedTraffic ←calcSavedTraffic(traffic, oldProb,
newProb) ×changeRate;
        if savedTraffic > adjustmentTraffic then adjustProb();
```

algorithm of a probability adjustment. Our results have shown that this still causes relatively frequent probability adjustments and thus too much control traffic. To compensate this, we add a fixed overhead to the calculated probability. The probability adjustment is performed periodically. For continuous events, this time may be chosen freely taking the desired accuracy and the traffic into account. For singular events, it must be ensured that the latencies of all events can be measured, i. e., each event must be transmitted at least twice. Hence, we calculate a value using the maximum supported data traffic inhomogeneity, the redundancy, and the desired latency of the event type. This ensures enough information to adjust the probability correctly.

Vehicle handling: For the propagation of the data collection probabilities, we use a publish-subscribe based geocast mechanism. Vehicles register to a topic that corresponds to a certain geo cell. Probability changes are propagated by publishing to the respective topic of the geo cell. This approach ensures that all vehicles are informed about the current collection probabilities.

VI. Client Side

Each event type that can be transmitted is assigned its own probability, i. e., the probabilities of event type may be different and individual for each geo cell. Using this approach, the server side does not need to know how many vehicles are actually driving in a certain geographic region, i. e., geo cell. The server side automatically adapts the probability considering the amount of transmissions for a certain event. The client side then transmits sensed event data according to the provided model.

The system is used to collect location based data by the use of mobile clients, i. e., vehicles serve as sensors. Collected information consists of singular events, e. g., the detection of traffic signs, as well as continuous events, e. g., the collection of temperature or rain drop rate. However, at the server side this information is used for a certain purpose. Participating clients should benefit from the gathered information and thus, this information has to be provided back to the vehicles. One possibility is to provide a service about information related to road segments in driving direction, a so called eHorizon. Based on this notion, we have introduced the concept of a remote eHoriozon in our previous work [24]. Such a remote eHorizon is a connected service that provides data about the current driving path. This data can consist of relatively static information about the road network serving as local map update or complement, transient static information like traffic signs, or even more dynamic information like the status of traffic lights, traffic status, accidents, or weather information. Such a technology provides the means for more predictive driver assistance systems, but is also able to optimize the data upload process of a system as described in the work at hand. By comparing sensed information with data provided within the eHorizon, the client can decide if information is relevant for transmission.

Figure 3 illustrates the decision process at the vehicle side. If sensed information is available, the first step is to decide with probability $p(e)$ if the data should be transmitted or discarded. In case of continuous events, i. e., a measurement task like temperature sensing, the data is then directly send. In case of singular events, it is checked if an according eHorizon information is available. If eHorizon information is available, data is only sent if this information is new or has changed, otherwise the data is sent directly. As extension an event can be triggered if an information that is available in the eHorizon data was not sensed. This will enable the system to unlearn existing knowledge, i. e., that a location based information does not exist anymore. An extension for sending continuous events would be a send on delta strategy that transmits the data only if a certain delta is exceeded. In any case, the server side requires a predefined redundancy to compensate false sensed information before accepting an information change in the database. However, the unlearning process and the eHorizon knowledge is out of scope for the following experimental evaluation.

VII. Experimental Results and Discussion

To evaluate ProbSense.KOM, the described model has been implemented in Java. The implementation consists of an implementation of the server side and an implementation for the vehicles. Since gaining experimental results using real vehicles was not possible due to the huge amount of required vehicles, we decided to run a simulation using SUMO [1].

To simulate realistic traffic, the TAPAS Cologne scenario was used, as mentioned before. This scenario is one of the largest available traffic scenarios for SUMO traffic simulator and consists of two hours of traffic in the metropolitan area

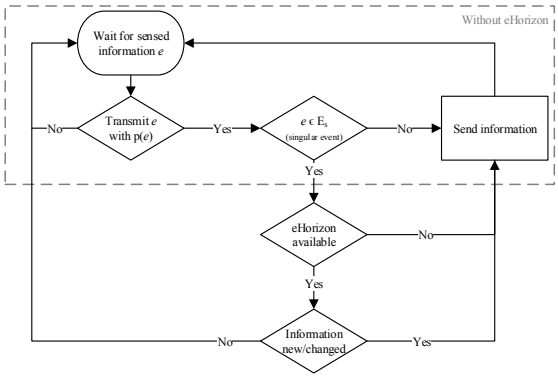

Figure 3: Overview of the client decision process.

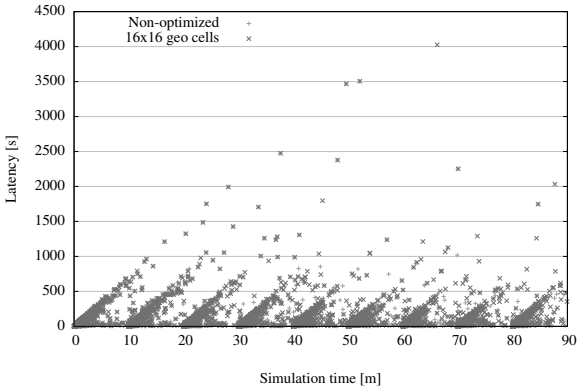

Figure 5: Reception latency distribution.

of Cologne. We skipped the first 30 minutes where the traffic is slowly increasing and used the following 90 minutes for our simulations. However, even this large scenario caps at a maximum of about 8,000 vehicles driving on the streets simultaneously on a road network of about $40 \times 40\ km^2$. Since ProbSense.KOM is only capable of reducing the data traffic if information is sent redundantly, the vehicle density was not satisfying. Therefore, we cloned each vehicle multiple times in the configuration file, which lead to more than $100,000$ vehicles on the streets simultaneously and about two million vehicles in total during the simulation time of two hours.

For both event types, one event each has been introduced: Traffic sign events are used for singular events, while temperature sensing events are used for continuous ones. Since the performance of this model is highly dependent on the desired data quality, the quality measures surely influence the simulation result. To be able to show the performance of our model in a realistic context, those quality measurements have been chosen as realistic as possible. We placed $2,000$ locations randomly to the considered road network that triggered a singular event on passing vehicles. The desired event reception redundancy was set to 5 and the latency was set to 10 minutes.

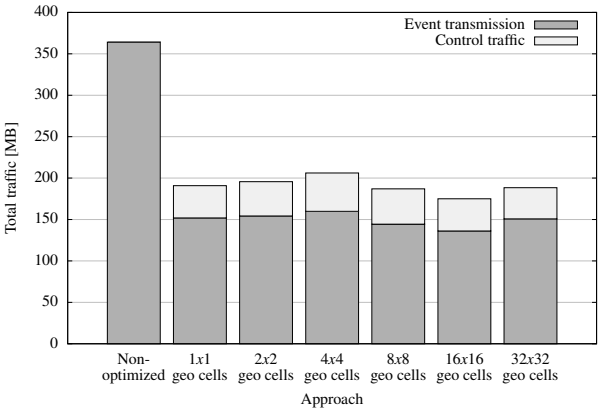

Figure 4: Total data traffic per approach.

The desired event reception rate for continuous events was set to 30 events per square kilometer and hour.

Since most of the events shall arrive in time, it is set that $99\,\%$ of all events shall be at the server in time. To evaluate the influence of the geo cell size, we simulated different geo cell patterns. First, we considered the the whole scenario as one single geo cell. Additionally, we divided the scenario into $2x2$, $4x4$, $8x8$, $16x16$ and $32x32$ geo cells, which results in geo cell edge lengths of about $40\ km$ down to about $1\ km$. For data serialization we have used the protobuf based sensor ingestion structure published by HERE [25]. The resulting data packet size was 123 Bytes for transmitted events and 15 Bytes for control data. All simulation runs, i. e., all geo cell pattern configurations, have been performed seven times. The results are the averaged results of the respective seven repetitions.

Evaluation results: The total data traffic of the different simulation settings is given in Figure 4. The non-optimized approach assumes that all events detected by the vehicles are directly sent. It can be seen that our approach is able to reduce the data traffic about $50\,\%$ compared to the non-optimized transmission approach. The amount of data traffic to propagate the calculated data collection probabilities consists of about $20\,\%$ of the total data traffic. The best result, i. e., the lowest total data traffic, was achieved using a $16x16$ grid. In this case, the map of the simulation scenario has been divided into $16x16$ squares that defined the geo cells. For each of these geo cells, the data transmission probabilities are calculated independently. The data reception latency stayed almost within the desired range of $600\,s$. Figure 5 shows the distribution of events arriving at a certain simulation time with a certain latency in one simulation execution of the $16x16$ geo cell configuration run. Most events with very long latencies above $600\,s$ are measured in the optimized and non-optimized approach synchronously. Therefore, in this situations, there are generally not enough vehicles traversing these events in order to store it at the server. For lower latencies, the resulting latencies are mostly above the reference values but still within the tolerated range of $600\,s$. Outlier above the $600\,s$ threshold that do not occur in the non-optimized approach are due to the

probabilistic behaviour of ProbSense.KOM, that only tries to get $99\,\%$ of all events in time. This results in a reduced amount of data traffic as depicted in Figure 4. Around $600\,s$, most event latencies differentiate between the two approaches, because the optimized approach lowers the transmission probability in order to lower the traffic. Therefore, cells with fewer vehicles had a higher transmission probability, while cells with plenty of vehicles had low transmission probabilities in order to decrease redundancy.

VIII. Summary & Conclusion

Within this work, we introduced our approach of a probabilistic data collection system. The purpose is to use vehicles as mobiles sensors to collect location based, or map based, data. Considered data of interest is divided into singular events, e. g., the detection of traffic signs, and continuous events, e. g., the collection of temperature sensing. Collected data is transmitted via a cellular link to a central server that is also responsible to manage the collection process. A probabilistic transmission model is used to reduce the amount of data traffic.

The server side calculates a transmission probability for each considered event type based on the incoming data rate. The scenario map is divided into geo cells, for which the calculation of the probabilities is conducted individually. The calculated transmission probabilities are propagated to the vehicle side, where these probabilities are used to decide if collected data should be transmitted. Thus, this transmission probability is used to manage the amount of transmitted data. The purpose is to reduce the amount of transmitted data while ensuring a defined quality gate with a probability of $99\,\%$. The quality of singular events is determined by the latency it takes to transmit a certain event to the server side with a defined redundancy.

In our evaluation, we set this redundancy to 5 and the tolerated latency to 10 minutes. For continuous events, the quality was set to 30 events per square kilometer and hour. If the incoming data rate is higher, then the server side reduces the transmission probability. If the incoming data rate is lower, then the transmission probability is increased respectively. With our approach, we were able to achieve a reduction of the total data traffic by about $50\,\%$ compared to a non-optimized approach, without harming data quality.

A further reduction of the total data traffic will be possible by a reduction of the defined redundancy or by increasing the tolerated latency. In our future work, we aim to combine our previously mentioned eHorizon service with the probabilistic data collection model. Using this model, we are confident to further improve the performance.

Acknowledgments

The work presented in this paper was partly funded by the German Federal Ministry of Education and Research (BMBF) under grant no. 01IS12054.

References

[1] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, December 2012.

[2] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A Survey of Mobile Phone Sensing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, 2010.

[3] H. Ma, D. Zhao, and P. Yuan, "Opportunities in Mobile Crowd Sensing," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 29–35, 2014.

[4] N. D. Lane, S. B. Eisenman, M. Musolesi, E. Miluzzo, and A. T. Campbell, "Urban Sensing Systems: Opportunistic or Participatory?" in *Proc. 9th Workshop on Mobile Computing Systems and Applications*, 2008.

[5] M. Shin, C. Cornelius, D. Peebles, A. Kapadia, D. Kotz, and N. Triandopoulos, "AnonySense: A System for Anonymous Opportunistic Sensing," *Pervasive and Mobile Computing*, vol. 7, no. 1, pp. 16–30, 2011.

[6] A. Hossain, P. Biswas, and S. Chakrabarti, "Sensing Models and its Impact on Network Coverage in Wireless Sensor Network," in *Proc. Third International Conference on Industrial and Information Systems*, 2008.

[7] K. Tischler and B. Hummel, "Enhanced Environmental Perception by Inter-Vehicle Data Exchange," in *Proc. IEEE Intelligent Vehicles Symposium 2005*, 2005.

[8] X. Yang, J. Liu, N. H. Vaidya, and F. Zhao, "A Vehicle-to-Vehicle Communication Protocol for Cooperative Collision Warning," in *Proc. First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, 2004.

[9] S. Vodopivec, J. Bešter, and A. Kos, "A Survey on Clustering Algorithms for Vehicular Ad-Hoc Networks," in *Proc. 2012 35th International Conference on Telecommunications and Signal Processing*, 2012.

[10] C. R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," *Selected Areas in Communications, IEEE Journal on*, vol. 15, no. 7, pp. 1265–1275, 1997.

[11] M. Gerla and J. T.-C. Tsai, "Multicluster, Mobile, Multimedia Radio Network," *Wireless Networks*, vol. 1, no. 3, pp. 255–265, 1995.

[12] M. Chatterjee, S. K. Das, and D. Turgut, "WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks," *Cluster Computing*, vol. 5, no. 2, pp. 193–204, 2002.

[13] P. Basu, N. Khan, and T. D. Little, "A Mobility Based Metric for Clustering in Mobile Ad Hoc Networks," in *Proc. 2001 International Conference on Distributed Computing Systems Workshop*, 2001.

[14] CONVERGE Project, "Deliverable D6 Final Assessment: Appendix 8.3 Results of Simulation," pp. 310–314, 2015. [Online]. Available: http://www.converge-online.de/doc/download/D6-AP8-Final-Assessment.pdf

[15] M. Miskowicz, "Send-on-Delta Concept: An Event-Based Data Reporting Strategy," *Sensors*, vol. 6, no. 1, pp. 49–63, 2006.

[16] Y. S. Suh, "Send-on-Delta Sensor Data Transmission With a Linear Predictor," *Sensors*, vol. 7, no. 4, pp. 537–547, 2007.

[17] D. Chu, A. Deshpande, J. M. Hellerstein, and W. Hong, "Approximate Data Collection in Sensor Networks Using Probabilistic Models," in *Proc. 22nd International Conference on Data Engineering*, 2006.

[18] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong, "Model-Driven Data Acquisition in Sensor Networks," in *Proc. Thirtieth International Conference on Very Large Data Bases*, 2004.

[19] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden, "CarTel: A Distributed Mobile Sensor Computing System," in *Proc. 4th International Conference on Embedded Networked Sensor Systems*, 2006.

[20] K. W. Chen, "CafNet: A Carry-and-Forward Delay-Tolerant Network," Ph.D. dissertation, Massachusetts Institute of Technology, 2007.

[21] S. Li, L. Da Xu, and X. Wang, "Compressed Sensing Signal and Data Acquisition in Wireless Sensor Networks and Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 9, no. 4, pp. 2177–2186, 2013.

[22] S. M. Stigler, "Gauss and the Invention of Least Squares," *The Annals of Statistics*, vol. 9, no. 3, pp. 465–474, 1981.

[23] G. Krucker, "Ausgleichs- und Interpolationsrechnung," 1998. [Online]. Available: http://www.krucker.ch/skripten-uebungen/IAMSkript/IAMKap3.pdf

[24] D. Burgstahler, A. Xhoga, C. Peusens, M. Moebus, D. Boehnstedt, and R. Steinmetz, "RemoteHorizon.KOM: Dynamic Cloud-based eHorizon," in *Proc. AmE 2016 - Automotive meets Electronics*, 2016.

[25] HERE, "Vehicle Sensor Data Cloud Ingestion Interface Specification v2.0.2," 2015. [Online]. Available: https://lts.cms.here.com/static-cloud-content/Company_Site/2015_06/Vehicle_Sensor_Data_Cloud_Ingestion_Interface_Specification.pdf

All online references in this paper were last accessed and validated in May 2016.