TUM

TECHNISCHE UNIVERSITÄT MÜNCHEN
INSTITUT FÜR INFORMATIK

# Beacon-based Vehicle Tracking in Vehicular Ad-hoc Networks

Karim Emara, Wolfgang Woerndl, Johann Schlichter

TUM-I1343

# Beacon-based Vehicle Tracking in Vehicular Ad-hoc Networks *

Karim Emara Wolfgang Woerndl Johann Schlichter
Department of Informatics
Technical University of Munich, Germany
*lastname@in.tum.de*

April 26, 2013

**Abstract**

Location privacy is one of the main challenges in vehicular ad hoc networks (VANET), which aims to protect vehicles from being tracked. Most of research work concerns changing pseudonyms efficiently to avoid linking messages through them. However, the sensitive information the vehicles send periodically in beacons make them vulnerable to tracking even if beacons are totally anonymous. On the other hand, vehicle tracking is useful in traffic efficiency and fleet management applications. In this paper, we used the nearest neighbor probabilistic data association (NNPDA) technique to track vehicles through information sent in anonymous beacons. We evaluated the implemented tracker against different vehicle densities, speeds, beacon rates, random noises and packet delivery ratios. The achieved tracking accuracy asserts the necessity of securing beacon messages from global observer attacks to be able to gain benefits of vehicle tracking.

## 1 Introduction

Vehicular ad hoc networks (VANET) emerge in past few years and gain a great interest in both research and industry for safety, traffic efficiency and infotainment applications. Most of the safety and traffic efficiency applications depend on *beacon* messages that vehicles broadcast periodically. These beacons usually contain a timestamp, a pseudonym, and the current vehicle position, speed and heading. By linking similar pseudonyms of subsequent beacons, vehicles can be tracked. To avoid this threat, previous work, as in [2, 3, 4], suggests to provide each vehicle with a set of pseudonyms where a different pseudonym is used every period of time according to a pseudonyms change policy [5, 6, 7]. However, beacons contain accurate information and are sent frequently (up to 10 Hz) which make vehicles vulnerable to tracking even if they change pseudonyms periodically. For example, Wiedersheim et al. [8] claim that vehicles sending beacon messages at 1 Hz and changing their pseudonyms every 10 seconds and having 20% penetration rate of all vehicles, an attacker can effectively track them with an accuracy of almost

*This technical report is the extended version of [1].

100%. They assumed a global attacker model which can eavesdrop all beacon messages sent in the network. This result means that even changing pseudonyms does not effectively protect the location privacy of vehicles due to the precise frequent information they send. Although revealed vehicles traces are anonymous, further correlation between real identities and those anonymous traces can be achieved as Golle and Partridge claimed in [9] which may lead users to reject VANET usage at all.

Besides privacy concerns, vehicle tracking is useful for important applications. For instance, it allows calculating average travel times of vehicles across individual roads [10]. This travel time is an excellent real-time indicator for traffic congestion which allows better route selection by drivers and better road management by road operator. Moreover, logistic companies use vehicle tracking for monitoring their fleet activity to analyze costs, plan for travel routes, measure the performance and improve the productivity [11]. Thus, it is desirable to gain vehicle tracking benefits without losing privacy.

In this paper, we develop and evaluate a vehicle tracker using anonymous beacons messages. We assume the same global attacker model used in [8] which eavesdrops every message sent in the network. It may be hard to have an external attacker who can cover the whole network, but this model is valid for compromised or corrupted authority through their deployed RSUs. On the other hand, we used a different tracking technique which is called Nearest Neighbor Probabilistic Data Association (NNPDA). The NNPDA is a simpler tracking algorithm than the Multi-Hypothesis Tracking (MHT) used in [8]. Its computational simplicity allows achieving real-time tracking even with dense networks and frequent beacons. Also, the vehicle model used in tracking includes the velocity and acceleration rather than using the position only. We assumed that the velocity and acceleration are already included in beacon messages based on the requirements of safety applications specified in [12]. We show that the tracking accuracy can be further enhanced even with different challenging environment settings and using totally anonymous beacons. The achieved tracking accuracy raises questions about the location privacy techniques and pseudonyms change policies deeply studied in the literature. It poses the need for securing and protecting not only the pseudonyms or real identities but also the vehicles information itself sent in beacons.

Next, we will discuss vehicle tracking and its components. In Section 3, the simulation scenarios and the experimental results are presented. In Section 4, we discuss our results and how they can be enhanced.

# 2 Vehicle Tracking

As stated previously, most of safety and traffic efficiency applications require vehicles to send its current position, speed, acceleration, heading and a pseudonym in beacon messages. Here, we assume that beacons are totally anonymous, and thus pseudonyms change policy is not a factor in the tracking vulnerability. By this assumption, vehicle tracking is considered to be a typical multiple target tracking (MTT) problem. The MTT is a well-studied problem and has comprehensive approaches and algorithms used in broad type of applications [13, 14]. It assumes a set of measurements or observations detected by a sensor periodically every time interval which is called a *scan*. Its goal is to find the best estimate of the targets state and the associated uncertainty in each scan. Measurements are assumed to be noisy and include clutter which are false detections not originated from real targets.

To explain the basics of MTT, let's start with the simplest case which is a single target tracking with no clutter. In this case, the sensor acquires a noisy measurement every scan and it is required to obtain the exact target state. Thus, a *state estimation* filter (e.g. Kalman filter) is used to obtain an accurate state using both the measurement gained from the sensor and a calculated state from a predefined kinematic model for that target. The estimation filter converges overtime to form a more accurate track for the target than that detected by the sensor.

When clutter presents, several measurements are detected in every scan but one of them is really originated from the target, if any. Thus, the estimation filter cannot be used directly as it is unknown which measurement belongs to the target. Thus, an association process is performed to identify which measurement is most likely originated from the target which is called *data association*. However, a validation process or *gating* is performed before that to avoid unnecessary computations. Gating aims to eliminate measurements that are less-likely to be originated from the target from being tested in the computationally intensive data association process. It forms a validation area around the track and excludes any measurement located outside this area from being tested in the data association.

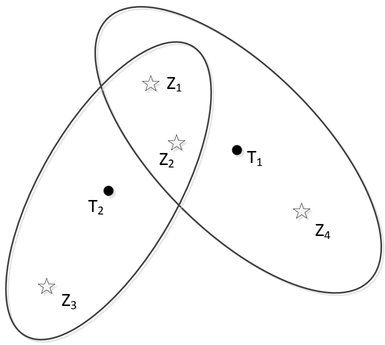



Figure 1: Gates of two tracks $T_1$ and $T_2$ with three measurements in each. Two measurements $Z_1$ and $Z_2$ located in the intersection of gates.

The multiple target tracking in clutter follows similar steps. First, let's assume that there are a set of tracks already established for the targets, then, a gate can be formulated around each track. As these gates can overlap together and measurements can be located in more than one gate as in Figure 1, the data association process for all tracks must be calculated together. Otherwise, the association will not be globally optimized leading to false assignments. If the number of targets are unknown and/or dynamic, a separate or joint process with data association handles track initiation, confirmation and deletion, which is called *track maintenance*. Figure 2 shows the main components of MTT.

Although vehicle tracking is an MTT problem, it has different goals, assumptions and constraints. First, vehicle tracking aims to link beacon messages originating from the same vehicle together forming an (anonymous) vehicle track. It does not aim to find the accurate vehicle state or to enhance vehicle measurements. Although it may use a state estimation filter, state estimation is not a goal by itself. Second, there is no clutter or false measurements assumed in beacon messages. All received messages reflect real targets with no doubt. Third, some of detection problems that may occur because of the limitation or deficiency of sensors are unlikely to occur in VANET beacons. Examples of these problems are the unresolved measurements problem, which occurs when a single measurement is formed from multiple targets and the multiple detection problem, which

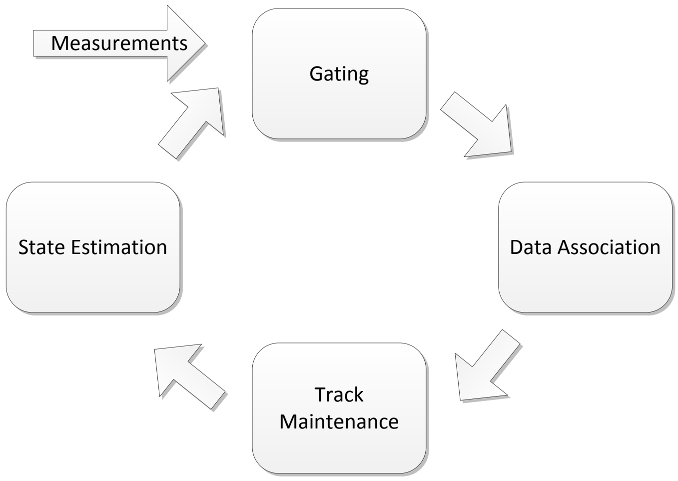



Figure 2: Components of multiple targets tracking

occurs when the same target is detected more than once in a single scan. These problems are considered to be the main challenges for data association [14]. Forth, the expected accuracy of broadcast information in beacons is better and the broadcast frequency is higher than those expected in MTT. This can be induced by the requirements of safety applications which many of them require precise location information with error less than one meter and high beacon rate of about 10 Hz [12]. Fifth, the vehicles movements, at the end, are predictable and constrained by roads and driving rules which leads to simpler vehicle modeling and tracking. All these differences between vehicle tracking and MTT propose that vehicle tracking can be accomplished effectively and efficiently using common even non-complex MTT approaches and will achieve an acceptable accuracy.

Next, we will discuss components of vehicle tracking in detail. Although vehicle tracking begins logically with gating, the state estimation is discussed first as the other components depend on it. Then, the other components are discussed in the logical order.

## 2.1 State Estimation

Vehicle state expresses the facts about the vehicle we are interested in, which may include the position, velocity and acceleration. Unfortunately, it is practically impossible to identify the exact vehicle state because the GPS receiver, speedometer, etc. are still sensors with limited precision and prune to imperfection and noise. Thus, to be able to track a vehicle and link its messages together, its exact state should be better estimated using a state estimation filter. The state estimation filter is not an interpolation or extrapolation but it gives a better estimate or correction for a state $x_k$ at time $k$ taking into account both the previous states $x_1$, $x_2$, $x_3$,...,$x_{k-1}$ and the inaccurate measurement $z_k$ detected at time $k$. The most common state estimation filter is the Kalman filter [15]. The Kalman filter (KF) is a set of mathematical equations that provide an efficient recursive method to estimate the state of a stochastic process, such that it minimizes the mean of the squared error. In order to use Kalman filter to estimate the vehicle state, vehicle dynamics should be modeled in accordance with the Kalman filter model. The basic Kalman filter assumes the underlying system to be linear dynamical system where the transition from the state at time $k$ to that of time $k+1$ is given by a linear equation. Also, it assumes that the process noise and the measurement noise have Gaussian distribution.

We model the vehicle motion process as a linear dynamic model with Gaussian-distributed noise defined as:

$$x_k = Ax_{k-1} + w \tag{1}$$

where $x_k$ is the vehicle state vector at time step $k$ and $A$ is the transition matrix that advances the state one step ahead. The random variable $w$ is the process noise with normal distribution $\mathcal{N}(0, Q)$ where $Q$ is its covariance matrix. The measurement $z_k$ of the state $x_k$ is computed as:

$$z_k = Hx_k + v \tag{2}$$

where $H$ is the model matrix that maps from the state space to the measurement space. The random variable $v$ is the measurement noise with normal distribution $\mathcal{N}(0, R)$ where $R$ is its covariance matrix. We assume that both $Q$ and $R$ do not change over time.

We defined the state vector $x_k$ to be position $p$, velocity $v$ and acceleration $a$ in the 3D Cartesian coordinate. The transition matrix $A$ is obtained using motion equations forming an 9x9 matrix. However, such large dimension of the state vector and the transition matrix leads to inefficiency in computations. Thus, as recommended in [13], the components of each coordinate are decoupled as they are independent from each other. Thus, the state vector and the transition matrix of each coordinate $x, y$ and $z$ are defined as follows:

$$x_i = \begin{bmatrix} p_i \\ v_i \\ a_i \end{bmatrix}, A_i = \begin{bmatrix} 1 & t & t^2/2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where the subscript $i$ refers to the $x, y$ or $z$ coordinate, and $t$ is the beacon time interval. The subscript $i$ is omitted later on for simplicity but it is worthy to note that any reference to the state vector $x$ means only a single part of the vector and it should be implicitly repeated three times, one for each coordinate. We assumed the beacon messages contain the current position, speed, acceleration and heading (i.e. cosine of thetas in each direction) based on the specifications of safety applications discussed in [12]. Authors of [8] worked on the position only which results to low tracking accuracy as we show in Section 4. We preprocessed this information before sending it to the tracker to avoid non-linearity in the model. We calculated the velocity and acceleration in each direction using the given heading thetas and the scalar values of speed and acceleration. Thus, the component of the measurement vector $z_k$ are similar to those of the state vector $x_k$. Therefore, the matrix $H$ is just an identity matrix.

$$z_k = \begin{bmatrix} p \\ v \\ a \end{bmatrix}, H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

Note that the measurement vector $z_k$ and the model matrix $H$ are again for a single coordinate similar to $x_k$ and $A$.

For the process noise, we assume $w = \begin{bmatrix} t^2/2 & t & 1 \end{bmatrix}^T \sigma_{as}^2$, where $\sigma_{as}^2$ is the acceleration variance in the process model. Thus, the covariance matrix $Q$ can be defined as:

$$Q = E(ww^T) = \begin{bmatrix} t^4/4 & t^3/2 & t^2/2 \\ t^3/2 & t^2 & t \\ t^2/2 & t & 1 \end{bmatrix} \sigma_{as}^4 \tag{5}$$

For the measurement noise, we assume the variances in measurements of position ($\sigma_p^2$), velocity ($\sigma_v^2$) and acceleration ($\sigma_{am}^2$) are given to the filter as parameters. Thus, the covariance matrix $R$ is defined as:

$$R = \begin{bmatrix} \sigma_p^2 & 0 & 0 \\ 0 & \sigma_v^2 & 0 \\ 0 & 0 & \sigma_{am}^2 \end{bmatrix} \tag{6}$$

Values of these parameters are carefully selected as discussed in Section 3.2. Now, the vehicle model is formed and can be used in Kalman filter as shown next.

The Kalman filter is a recursive algorithm and circulates between predication and update phases. At time step $k$, the predication phase calculates a predicated (a priori) state estimate $\hat{x}_k^-$ using the estimated state $\hat{x}_{k-1}$ of the previous time step $k-1$. It also calculates a predicated (a priori) error covariance matrix $P_k^-$ which indicates the accuracy of the predicated estimate as specified in (7). The predicated state estimate $\hat{x}_k^-$ is called also *a priori* because it does not include the measurement of the current time step yet.

**Predication Phase:**

$$\begin{aligned} \hat{x}_k^- &= A\hat{x}_{k-1} \\ P_k^- &= AP_{k-1}A^T + Q \end{aligned} \tag{7}$$

where $A$ and $Q$ are matrices defined in (3) and (5), respectively. For the initial state vector $\hat{x}_0$, it is assumed that the measurements of the first scan form them and form the initial tracks. Also, the initial error covariance matrix $P_0$ is formed to have a parametric error in position while zero error in velocity and acceleration as follows:

$$P_0 = \begin{bmatrix} p_0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{8}$$

where $p_0$ is a parameter given to the filter.

The update phase calculates the Kalman gain $K$ to update the predicated estimate by the observed measurement at the current step. Also, it computes the residual or innovation $\tilde{z}_k$ which is the difference between the actual measurement and the estimated one and the innovation covariance matrix $S$ which indicates the accuracy of the residual. Both the residual $\tilde{z}_k$ and its covariance matrix $S$ are used later in the gating component.

**Update Phase:**

$$\begin{aligned} S &= HP_k^-H^T + R \\ K &= P_k^-H^TS^{-1} \\ \tilde{z}_k &= z_k - H\hat{x}_k^- \\ \hat{x}_k &= \hat{x}_k^- + K\tilde{z}_k \\ P_k &= (I - KH)P_k^- \end{aligned} \tag{9}$$

where $H$ and $R$ are matrices defined in (4) and (6), respectively, and $I$ is the identity matrix. More details about Kalman filter and its derivations can be found in [16].

## 2.2 Gating

Assuming a track is established for each vehicle, a measurement-to-track association should be performed to assign each measurement to the correct track. Before that, a gating process is required to narrow the association scope and eliminate measurements that are less likely to be assigned to each track. The most common gating technique is the ellipsoidal gate. The ellipsoidal shape is a consequence of the assumption that the error in the residual ($\tilde{z}_k$) is Gaussian [17]. The ellipsoidal gating defines a gate $G$ such that the association is allowed if the norm of the residual vector ($d^2$) is within this gate $G$:

$$d^2 = \tilde{z}^T S^{-1} \tilde{z} \leq G \tag{10}$$

where $\tilde{z}$ and $S$ are the residual vector and its covariance matrix respectively, defined in (9). The norm $d^2$ is calculated for all combinations of measurements and tracks. When a measurement satisfies the gating inequality with a track, it is kept as a validated measurement for that track. Otherwise it will be excluded from the possible assignments in data association. On the other hand, the gate size $G$ can be calculated adaptively based on the probability of detection $P_D$ and the residual vector. The probability of detection can be envisioned as the packet delivery ratio expected in vehicular network. However, as stated in [13], $d^2$ is assumed to have Chi distribution $\chi^2_M$ where M is the degree of freedom or the dimension of the measurement vector. For the model specified in the previous section ($M = 9$), $G$ is set to be more than 22.

## 2.3 Data Association

After measurements are validated for each track, it is likely to have measurements in more than one gate, as illustrated in Figure 1. As it is not allowed to assign a measurement to multiple tracks, it is necessary to do association for all tracks together to avoid incorrect or sub-optimal solutions. There are several association approaches and differ in how the assignment is accomplished. Some approaches, such as the global nearest neighbor (GNN), find the best measurement to update each track. However, there are others, such as joint probabilistic data association (JPDA), incorporate several measurements with weighting probabilities to update a single track. Also, the assignment decision can be based only on the measurements of the current scan or can be postponed several scans until finding the best hypothesis, as in Multi-hypothesis tracking (MHT).

The GNN is the simplest data association approach as it handles the association problem in straightforward way. It calculates a cost for each measurement-to-track assignment forming an assignment matrix. Then, it uses an efficient method for solving the assignment problem, such as Auction algorithm [18], to find the maximum number of possible assignments which also minimizes the total cost. The cost function can be defined in multiple ways, one of them is to define a statistical distance for the assignment of measurement $j$ to track $i$ as:

$$d^2_{G_{ij}} = d^2_{ij} + ln(|S_{ij}|) \tag{11}$$

where $d^2_{ij}$ is as defined in (10) and $ln(|S_{ij}|)$ is the logarithm of the determinant of the innovation covariance matrix $S_{ij}$ defined in (9). This last term is used to penalize tracks with great uncertainty expressed in large innovation matrix. There are several approaches that enhance the association of GNN such as branching to multiple hypotheses or calculating the cost function using subsequent scans. However, the GNN becomes obsolete because of the feasibility of the advanced techniques, such as JPDA and MHT [13].

The JPDA updates the track with a weighted average of all the measurements within its gate. The weighting function for assigning a measurement to a track can be calculated as follows. For each scan, one calculates the probability of each hypothesis that assigns each validated measurement to each track. Then the probability of a particular measurement-to-track association is calculated by the sum of probabilities of all hypotheses which include such association. Unfortunately, this method is not suitable for vehicle tracking. First, it is evaluated in literature, as in [19], to be not suitable for closely spaced targets which is highly expected on roads. Second, the idea of updating a single track by multiple measurements is not logical in vehicle tracking, because it is guaranteed that different measurements or beacons necessarily correspond to different vehicles. Thus, it is not logical to update a vehicle track by information of other vehicles as it definitely results to deviation in the generated tracks. On the other hand, JPDA complexity is combinatorial as it requires generating all association hypotheses.

There is another simplified form of JPDA proposed in [19] which is called nearest neighbor PDA (NNPDA). It aims to fasten the association calculations and avoid weighted-average updating feature in JPDA. It calculates a probability for each measurement to track association, as in JPDA, but without generating the association hypotheses. Then it forms an assignment matrix with those probabilities and uses an assignment algorithm to select the optimal assignments. The probability $P_{ij}$ of assigning a measurement $j$ to track $i$ is defined as:

$$P_{ij} = \frac{G_{ij}}{T_i + M_j - G_{ij}}, \; G_{ij} = \frac{e^{-d_{ij}^2/2}}{(2\pi)^{N_m/2}\sqrt{|S_i|}} \tag{12}$$

where $G_{ij}$ is the Gaussian likelihood function associated with the assignment of measurement $j$ to track $i$, $T_i$ is the sum of likelihood functions $G_{ij}$ of track $i$, $M_j$ is the sum of likelihood functions $G_{ij}$ of measurement $j$. The $d_{ij}^2$ is the normalized distance between the measurement $j$ and track $i$ defined in (10) and the $|S_i|$ is the determinant of the residual covariance matrix defined in (9). The $N_m$ is the dimension of the measurement vector. After calculating all probabilities, an assignment matrix is formed to find the optimal associations that maximize the sum of probabilities. These optimal associations are used to update each track individually.

The MHT is different from GNN and PDA approaches in that it postpones the association decision for multiple subsequent scans. It generates hypotheses for all validated measurements with each tracker like JPDA but it propagates (a subset of) them for subsequent scans aiming to resolve the uncertainty. Surely propagation of hypotheses over scans leads to combinatorial explosion, thus multiple techniques are used to avoid this using pruning, clustering or track merging.

The choice of the right data association approach is crucial and depends on the application specifications and requirements. In general, the data association accuracy is affected by the distance between vehicles and the beacon time interval [13]. Largely spaced vehicles and shorter beacon time intervals pose less instability in the association. However, simple association approaches may not enhance the stability even if the time interval is decreased [13]. Thus, sophisticated techniques should be used even with good system conditions. We used the NNPDA technique for data association. We aim to evaluate another approach rather than MHT already evaluated in [8]. Also, we think that the NNPDA is simpler than MHT which allows real-time calculations even with large number of vehicles.

## 2.4 Track Maintenance

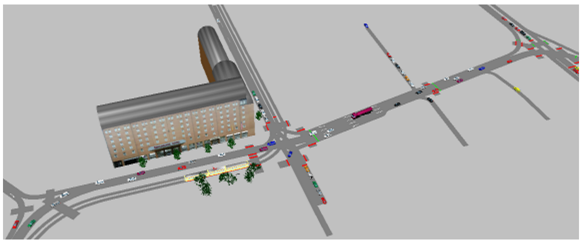
In MTT, a track maintenance logic is required to initiate, confirm and delete tracks. When a measurement is received and not assigned to a previously established track, a new track is initiated. However, this measurement may be a false alarm, thus this track is considered as a *tentative* track until it is confirmed in subsequent scans. The track confirmation can be typically done if $M$ correlating measurements received in $N$ scans and assigned to this track. Another approach is to define a score function for tentative tracks and confirm them once they exceed a predefined threshold. On the other hand, when a track is not updated for a while, it should be deleted to avoid further wrong associations and eliminate computational overhead. A typical deletion rule is to delete a track after a deletion tolerance interval of $N$ consecutive scans with no update. Also, a score function can be used for this purpose.

In vehicle tracking, there is no assumed clutter at all. Thus, the track maintenance is simpler than in MTT. For example, we assumed that a track is initiated and confirmed immediately once a measurement is received and not assigned to a previously established track. For track deletion, we hold the track for two consecutive scans with no update, and it is deleted after that. However, we think that this deletion tolerance interval should be modified with respect to the expected packet delivery ratio in vehicular network. If it is small and multiple beacons are lost in sequence, then the track will be deleted quickly causing several discontinuities in the vehicle trace. Because if it is large, different vehicle traces may be merged or joined into a single track. Thus, this parameter should be well-selected to avoid such cases as discussed in Section 3.4.

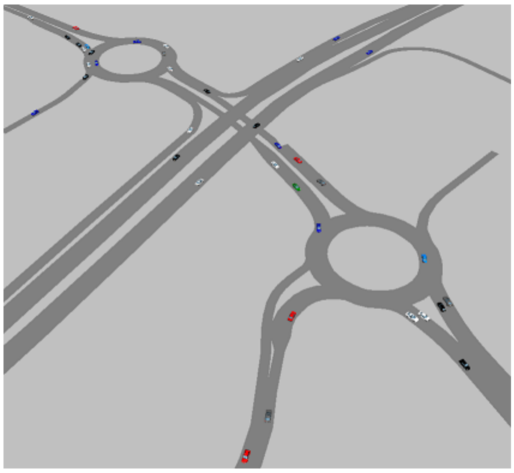
# 3 Evaluation

The developed vehicle tracker is evaluated using the commercial VISSIM traffic simulator. VISSIM is a microscopic, time-step and behavior-based simulation to model vehicle traffic and public transport operations. Its traffic simulator is a microscopic traffic flow simulation model including the car following and lane change logic [20]. VISSIM uses a psycho-physical driver behavior model developed in [21]. The basic concept of this model is that the driver of a faster vehicle starts to decelerate as she reaches her individual perception threshold to a slower vehicle. Since she cannot exactly determine the speed of that vehicle, her speed will fall below that vehicle's speed until she starts to slightly accelerate again after reaching another perception threshold. This results in an iterative process of acceleration and deceleration. VISSIM supports also great control on the road network and traffic customization. It supports drawing roads and connection links between them, adding priority rules, stop signs and traffic lights. It allows traffic composition of several vehicle types and characteristics. It allows specifying traffic entering rate, vehicle desired speeds and routes decisions. VISSIM has 2D and 3D graphical real-time display and supports information logging on a discrete time basis down to 100 ms. We used VISSIM for its realistic mobility model and variety of parameters which allow generating realistic vehicle traces.

We used the logging feature to generate vehicles states information every 100 ms in a trace file. Such trace file includes the position in the three coordinates, scalar values of speed and acceleration, along with the vehicle ID which is used only in evaluation. The vehicle heading is not directly generated from VISSIM, therefore it is calculated using the vehicle position in the next time step. At last, the velocity and acceleration vectors are calculated for each coordinate. Thus, the final trace file passed to the tracker contains the position, velocity and acceleration in the three coordinates along with vehicle ID and grouped by time step.



(a) Urban Network



(b) Highway Network

Figure 3: The main parts of road networks of the simulation scenarios

## 3.1 Scenarios and Simulation Setup

We choose two scenarios to evaluate the vehicle tracker: urban and highway road networks. Networks of both scenarios are included in VISSIM demos. As shown in Figure 3a, the urban scenario is a part of roads in Luxembourg city and consists of three intersections controlled by fixed-time traffic lights along with five join and exit roads. The main road is multi-lane single direction and is crossed by two-direction single-lane roads. The network size is about 850 m by 500 m. The Figure 3b shows the highway scenario which consists of a multi-lane two-direction main road with two roundabouts and a bridge passes over it. As this network represents a highway, there is no traffic lights or stop signs. The network size is about 550 m by 500 m. For both scenarios, the simulation duration is 300 seconds which is sufficient for traffic to enter and exit the network several times with all different routes. The decision routes that vehicles follow are preconfigured in the VISSIM network file and used as they are.

For evaluation, we defined a set of parameters to test the tracking accuracy against them. These parameters are summarized in Table 1 for urban and highway scenarios showing their value ranges. In addition, Table 1 shows the common value which is assigned to this parameter when other one or two parameters are varying. The first parameter is the vehicle entrance rate to the network which indicates the vehicle density. Generally, an entrance point is located in the starting point of each road in the network. The arrival rates are chosen to avoid frequent long traffic jams. Such rates result in a maximum number of simultaneous vehicles 25-195 vehicles in urban scenario and 20-64 vehicles in the highway scenario. The second parameter is the desired speed that the drivers want to reach. In VISSIM, the desired speed is not a fixed value for all vehicles but it is distributed around the specified value. Also, it is not necessary for vehicles to drive in such speed constantly, however, their actual speed depends on the traffic and the logic of the mobility model. According to the VSC report [12], most of safety applications require a minimum update frequency between 1 to 10 Hz. Thus, beacon time interval is chosen to start from 0.1 second to 5 seconds to evaluate longer time intervals may be used by applications in future. The vehicle position and speed retrieved from VISSIM is perfectly measured where it is not the case in reality. Thus, a normally distributed random noise is always added to the position and speed values. Several noise distributions

| Parameter | Urban Range | Urban Common | Highway Range | Highway Common |
|---|---|---|---|---|
| Entrance Rate (Vehicle/hour) | 100 - 600 | 300 | 300 - 1000 | 600 |
| Max simultaneous vehicles | 25 - 195 | 77 | 20 - 64 | 35 |
| Desired Speed (km/h) | 30 - 70 | 50 | 80 - 130 | 100 |
| Beacon Interval (s) | 0.1 - 5 | 0.5 | 0.1 - 5 | 0.5 |
| Position Noise (m) | $\mathcal{N}(0, 0-5)$ | $\mathcal{N}(0, 1)$ | $\mathcal{N}(0, 0-3)$ | $\mathcal{N}(0, 0.5)$ |
| Speed Noise (km/h) | $\mathcal{N}(0, 0-3.5)$ | $\mathcal{N}(0, 2)$ | $\mathcal{N}(0, 0-6.5)$ | $\mathcal{N}(0, 3.5)$ |
| Simulation Time (s) | 300 | | | |
| Simulation Runs | 10 (for each experiment) | | | |

Table 1: Simulation parameters in urban and highway scenarios

are evaluated to determine the noise impact on the tracking accuracy. For the position noise, the common standard deviation is chosen to be slightly larger in the urban scenario as the the GPS receivers are prune to larger localization errors within buildings than in open areas of highways. For the speed noise, we used the half of 10% error margin allowed by the authority (i.e. tolerance drift in the speed meter) as a standard deviation of the random noise. We run the simulations 10 times for each experiment and taking the average tracking accuracy as a result. Simulation runs are similar to each other in vehicle traces but the random noises added to position and speed are different resulting to different tracking scenarios which in turn lead to different results.

### 3.2 Parameters Selection

Parameters of Kalman filter and Gating should be adequately selected as they greatly influence the tracking results. We chose a simple case from each scenario and tried all parameters combinations several times with noisy positions and several beacon time intervals. Then, for each parameter, we tried to find a smaller that results in the best tracking accuracy on average. We repeated this procedure again with this smaller optimized range but with fine stepping to obtain a well-tuned value for each parameter. Table 2 shows the tested ranges for each parameter and its value optimized for each scenario.

| Parameter | Test Range | Urban | Highway |
|---|---|---|---|
| **Kalman filter:** | | | |
| $p_0$ | 20 - 70 | 50 | 50 |
| $\sigma_{as}^4$ | 0.1 - 5 | 0.7 | 5 |
| $\sigma_p^2$ | 1 - 25 | 5 | 2 |
| $\sigma_v^2$ | 0.5 - 5 | 2 | 5 |
| $\sigma_{am}^4$ | 0.5 - 9 | 1 | 7 |
| **Gate size $G$:** | 20 - 70 | 30 | 30 |

Table 2: Kalman filter and gating parameters

### 3.3 Experimental Results

In our evaluation, we used the maximum continuous tracking period percentage as a metric for tracking accuracy. To explain how this metric is calculated, we show first how the tracker practically works. Initially, the tracker creates a set of tracks for beacons appear in the first scan. Then, it assigns beacons in subsequent time steps to the established tracks and may start new tracks. However, it may mix vehicle traces by mistake and assign a beacon to a wrong track. Later, it can overcome this wrong interruption and return assigning beacons to the original correct track. Our metric expresses the maximum continuous period of assigning beacons of a specific vehicle to a track with no interruption divided by the total period of this vehicle appeared in the simulation, averaged over all vehicles. For example, assume a vehicle appeared in the simulation for 10 time steps, and thus it generates 10 beacons. Assume that the tracker assigned the first three consecutive beacons to track A, the next two beacons to track B as it assumed they belong to another vehicle and finally the last five beacons to track A again, as shown in Figure 4. Thus, the maximum continuous tracking period percentage is the period of the third tracking segment which is five time steps or beacons divided by the vehicle total lifetime which is ten steps resulting to 50%. This metric is calculated for each vehicle and averaged over all vehicles to obtain the tracking accuracy of a simulation run. This metric is similar to the one used in [8] except that they allow a single interruption in the calculated period.

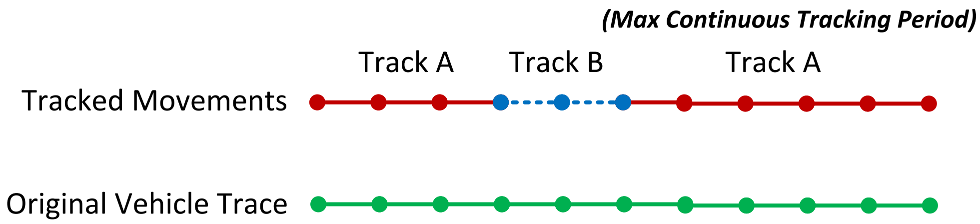
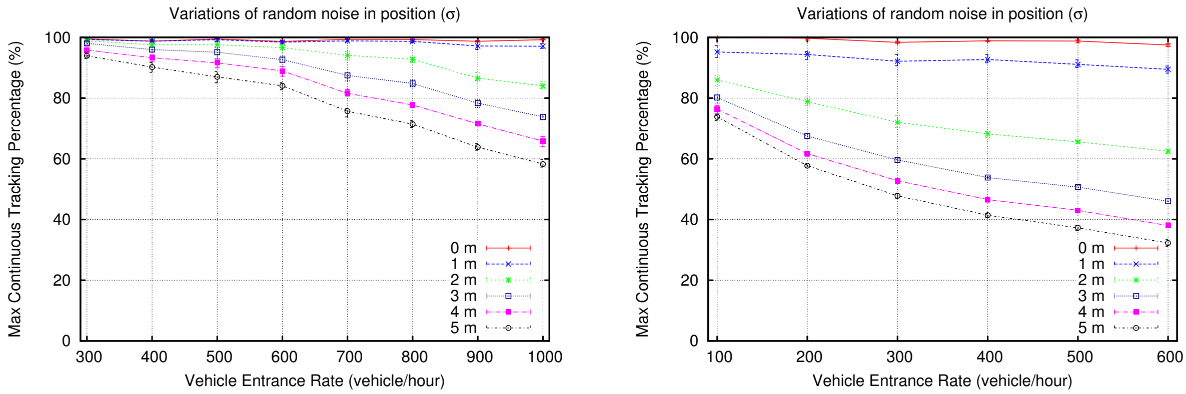



Figure 4: An illustration for calculating the maximum continuous tracking period metric for one vehicle

Before we discuss individual experimental results, general observations will be noted. In general, the tracking accuracy in highway scenarios is better than those in urban scenarios. This observation is expected, because, in the highway scenario, there is no traffic light or stop sign and vehicles travel in high speeds which leads to large separation distance between vehicles, although the entrance rates are higher than those in urban scenario. This confirms concepts discussed in Section 2.3 that the largely separated targets give better stability in data association. Second, the error bars drawn in graphs almost do not appear, which induces the stability of NNPDA algorithm in vehicle tracking against random noises in position and speed.

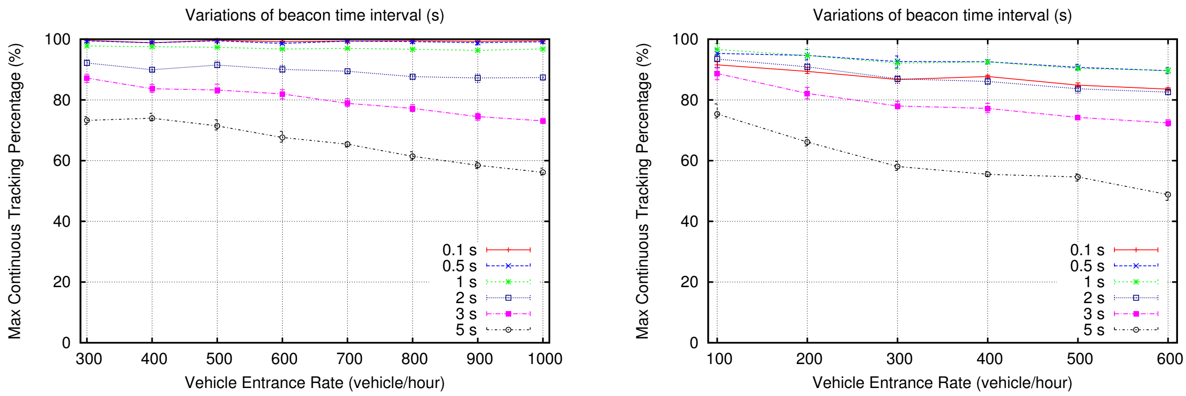
The first evaluation tests the tracking accuracy versus different vehicle entrance rates with variant random noises in position, as shown in Figure 5. It can be shown that the tracking is accurate (more than 90%) regardless the entrance rate for less noisy positions ($\sigma < 2$ meters) for both scenarios. This means that the positioning accuracy requirement of safety applications such as lane change and forward collision detection applications make vehicles lose their location privacy, regardless the entrance rate or vehicle density. In case of more noise, the vehicle entrance rate becomes a factor and the tracker is more confused in beacons associations resulting to lower accuracy. However, the impact of entrance rate in urban scenario is greater than that in highway scenario because the separation distances between vehicles are smaller in urban scenario. This means that for low vehicle densities or largely spaced vehicles, a high tracking accuracy can be achieved even with large random noises.

Next, we evaluate the effect of beacons time intervals with different entrance rates as




(a) Highway scenario (b) Urban scenario

Figure 5: Vehicle density versus variations of random noise in position

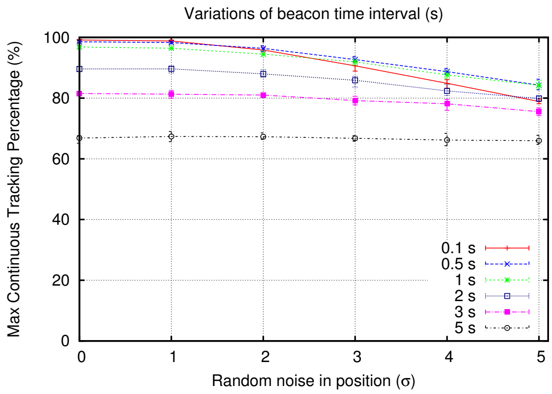



(a) Highway scenario (b) Urban scenario

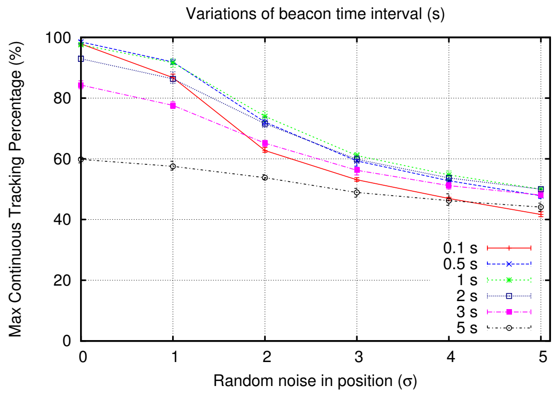
Figure 6: Vehicle density versus variations of beacon time intervals

shown in Figure 6. It is worthy to note that a normally distributed noise is still added to the position and speed in beacons as specified in the common value column in Table 1. First, in urban scenario, we notice that the tracking accuracy of the 0.5 and 1 second are better than that of the 0.1 second, regardless the entrance rate. Such unexpected accuracy reduction in the 0.1 second case occurs because vehicles positions are near to each other in subsequent time steps. Therefore, after adding the random noise, positions become more confusing to the tracker. On the other hand, the beacon time intervals upto two seconds generally achieve high tracking accuracy of 90% in highway scenario and 80% in urban scenario with little effect of vehicle entrance rate. In larger beacon time intervals (more than 2 seconds), the tracking accuracy decreases linearly with the increase of the entrance rate. This finding emphasizes the trade-off between safety applications requirements of 10 Hz or even 1 Hz beacons rate and location privacy.

From evaluations up till now, the impact of random noise in position and the beacon time interval on the tracking accuracy can be noticed. Therefore, we evaluate if these both factors are correlated, as shown in Figure 7. As the vehicle entrance rate is fixed in this case for each scenario, we can assume highway and urban scenarios represent low and high densities, respectively. For highway scenario, different position random noises



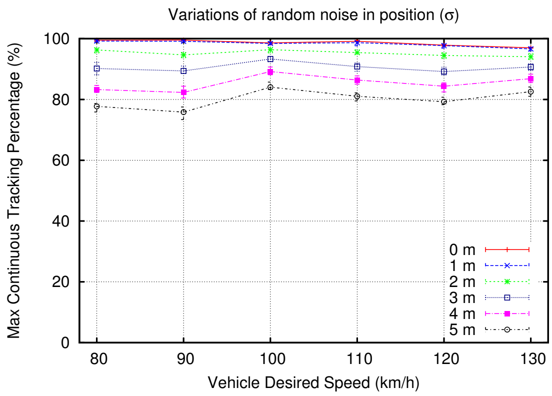

(a) Highway scenario



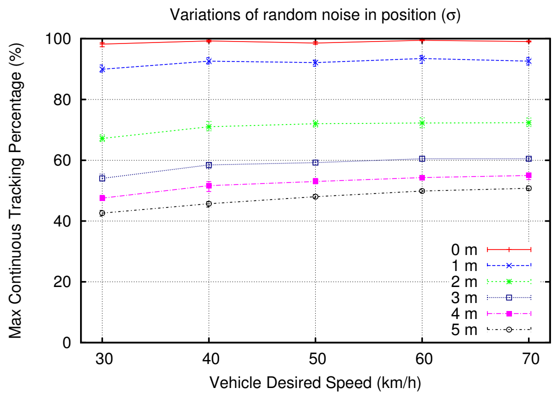

(b) Urban scenario

Figure 7: Random noise in position versus variations of beacon time intervals

do not affect the accuracy for large beacon intervals (more than 2 seconds), however it has a larger negative effect for smaller intervals, specially for the 0.1 second. On contrast, in the urban scenario, position noise greatly reduces the accuracy for all beacon intervals. Such different effect in both scenarios is expected as the tracker is confused more in the dense network than in sparse one when noise presents. Also, the achieved accuracy of different beacon intervals with large noises ($\sigma \geq 3$ meters) is almost near each other in urban scenario. This means that, in intermediate vehicle density, the accuracy is not greatly enhanced by using more frequent beacons when large position noise presents.



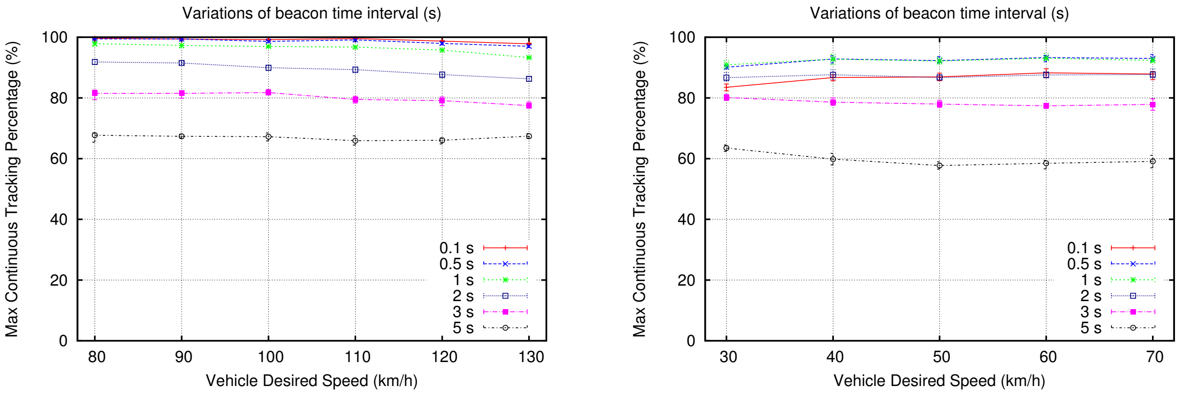

(a) Highway scenario




(b) Urban scenario

Figure 8: Vehicle desired speed versus variations of random noise in position

The next two evaluations test the effect of the vehicle desired speed on tracking accuracy versus random noises and beacon time intervals. Figure 8 shows that the accuracy does not change a lot with different speeds. It slightly increases with higher speeds in the urban scenario, while it oscillates slightly in the highway scenario. On the other hand, the impact of noise is greater in urban scenario which has lower speed range and denser than those in highway scenario. This happens as the separation distances between vehicles in the low-density highway scenario increase with higher speeds, and thus the tracker does not confuse greatly even with larger noise. However, in the dense urban scenario, separation distances may increase in higher speeds but they are still insufficient for discrimination by the tracker when large noises are added. This results in decreasing the accuracy, in general, for larger noises.




(a) Highway scenario

(b) Urban scenario

Figure 9: Vehicle desired speed versus variations of beacon time intervals

In Figure 9, the last evaluation is presented. It shows variations of the beacon time intervals versus the vehicle desired speed. In general, it can be inferred that the desired speed does not affect the accuracy when other factors such as vehicles density, position noise and beacon time interval are fixed. On the other hand, the accuracy of the beacon time interval of 0.1 second is lower than that of larger intervals (up to 2 seconds) because the added noise in closely separated vehicles confuses the tracker more than in sparse environments, as explained earlier.

The maximum continuous tracking period metric represents the quality of tracking by showing how much of the vehicle trace can be tracked. However, it does not show how many vehicles are perfectly tracked. For example, the tracker can track on average 50% of vehicle traces but in the same time there are so many vehicles are still perfectly tracked. Thus, we use an additional metric to clarify such cases which is the percentage of vehicles that are perfectly tracked. We assume that the perfectly tracked vehicles are those vehicles tracked continuously for more than 98% of their original trace without any interruption. We left 2% as a tolerance for track initiation. Over thousands of simulation runs performed, we show the relation between those two metrics as shown in Figure 10. This figure shows the average and range of percentages of perfectly tracked vehicles versus the used metric so far for both urban and highway scenarios. It shows that the possibility of perfect tracking for many vehicles still exists even with low average tracking accuracy. For example, 40% of vehicles can be perfectly tracked on average in urban scenario with tracking accuracy of only 70%. Also, about 60% of vehicles are perfectly tracked on average for tracking accuracy of 85%. This means that even with conditions resulting to intermediate tracking accuracy, many vehicles can be perfectly tracked on average and totally losing their location privacy. Interestingly noted from Figure 10, the average of perfectly tracked vehicles is more in urban scenario than in highway scenario.

## 3.4 Packet Delivery Ratio Effect

In previous evaluations, a global perfect attacker is assumed who can eavesdrop every message sent to the network. However, this model is not realistic due to the typical limitations of wireless communication such as packet loss. Packet loss is common in

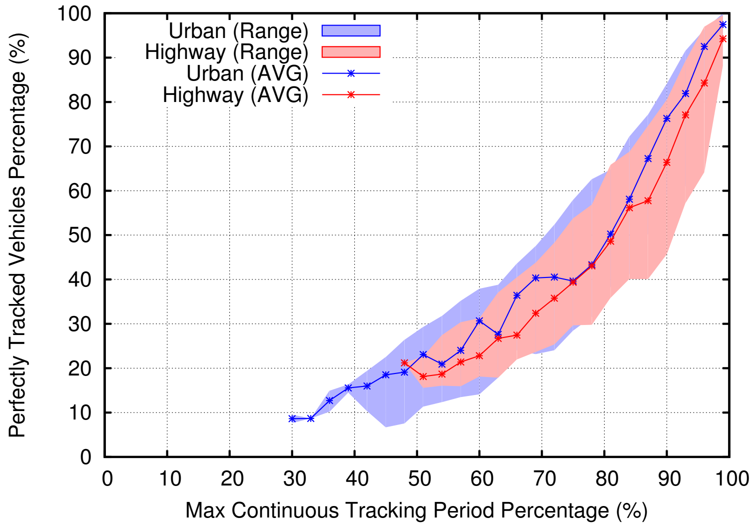


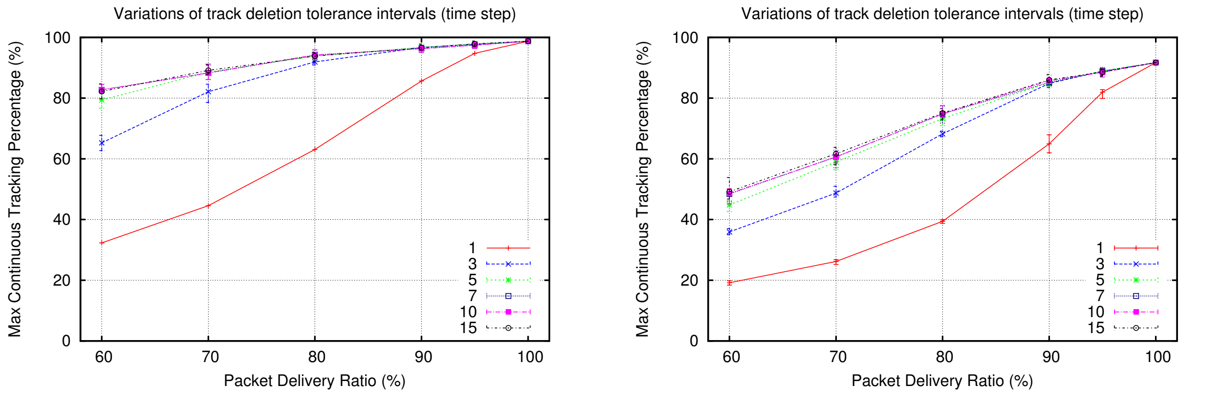

Figure 10: Percentage of continuous tracking period versus percentage of perfectly tracked vehicles

wireless communication due to several reasons such as signal degradation and channel congestion. The effect of packet loss on vehicle tracking is that random beacons may be lost every time step and thus the tracker may be more confused due to lost beacons. However, before we evaluate such effect, we need to study what is the suitable value for track deletion tolerance interval discussed in Section 2.4 as we think it is related to the packet loss ratio.

We modified our implementation so that it skips random beacons every time step based on the given packet delivery ratio. Both urban and highway scenarios are examined with a range of packet delivery ratios between 60% and 100% and several deletion tolerance intervals from 1 to 15 time steps. A deletion tolerance interval of one time step means the track is deleted if it is not updated for two consecutive time steps and so on. We run simulation using the parameters common values specified in Tables 1 and 2 except the gate size $G$. We find that the gate size should be adaptively selected according to the packet delivery ratio which refers to the probability of detection $P_D$ in the MTT terms. We set the $G$ to be the inverse of Chi distribution $\chi^2_M$ of the expected packet delivery ratio.

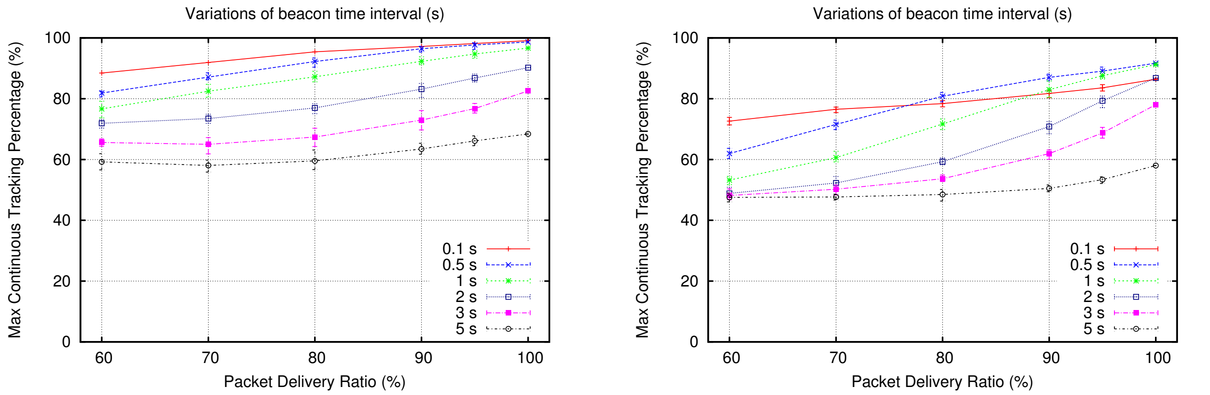
As shown in Figure 11, the deletion tolerance interval does not play any role in the case of the perfect packet delivery (100%). This is important as our previous results assumes a tolerance interval of two time steps and perfect packet delivery ratio, thus, we do not need to repeat the previous experiments. However, for lower packet delivery ratios, the deletion tolerance interval decreases the tracking accuracy specially for intervals smaller than or equal five time steps. Also, larger tolerance intervals do not enhance results already degraded by packet loss, they almost achieve the same accuracy. Thus, we can say that low values of deletion tolerance intervals may decrease the tracking accuracy but the higher ones do not enhance it. On the other hand, the tracking accuracy is seriously degraded in urban scenario than in highway scenario for lower packet delivery ratios ($<$ 90%). Because of the dense environment, closely spaced vehicles and larger positioning noise of the urban scenario, the tracker is more confused and wrongly assigned beacons of some vehicles to ones which missed their beacons.

Furthermore, we evaluate the effect of the packet delivery ratio with respect to the




(a) Highway scenario (b) Urban scenario

Figure 11: Packet delivery ratio versus variations of track deletion tolerance intervals



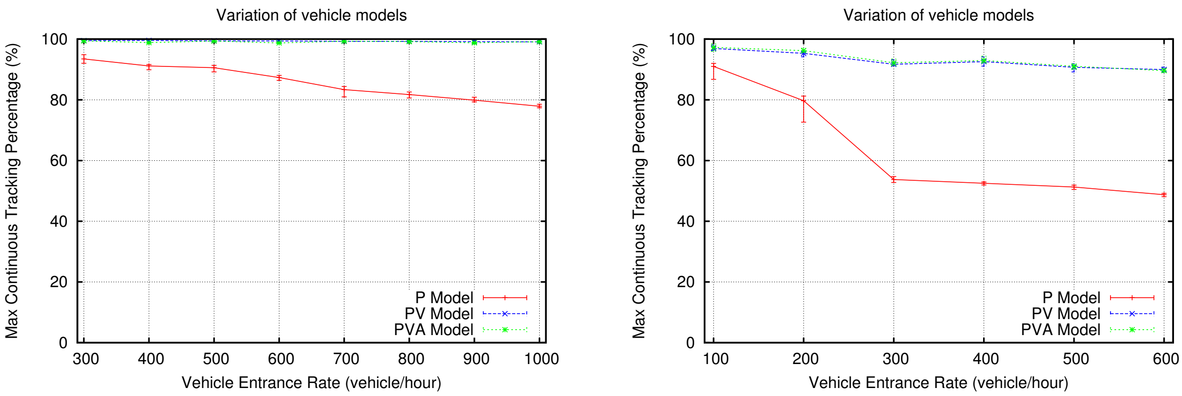

(a) Highway scenario (b) Urban scenario

Figure 12: Packet delivery ratio versus variations of beacon time intervals

beacon time interval as shown in Figure 12. Based on the previous result, we choose the track deletion tolerance interval to be ten time steps. For the highway scenario, the tracking accuracy is reduced linearly in small beacon intervals (less than or equal 1 second). However, the accuracy becomes a constant with lower packet delivery ratios (less than or equal 80%) in larger beacon intervals. On the other hand, in urban scenario, the accuracy is degraded greatly in all intervals (except in 5 seconds interval) when decreasing the packet delivery ratios. However, for 0.1 second beacon time interval, the accuracy is reduced only by 15% from the highest to lowest packet delivery ratios. But in the other beacon intervals, the accuracy is degraded by more than 30% except the beacon interval of 5 seconds. In other words, the accuracy reduction caused by the packet loss can be mitigated in sparse environment as in highway scenario and using short beacon time intervals.

# 4 Discussion

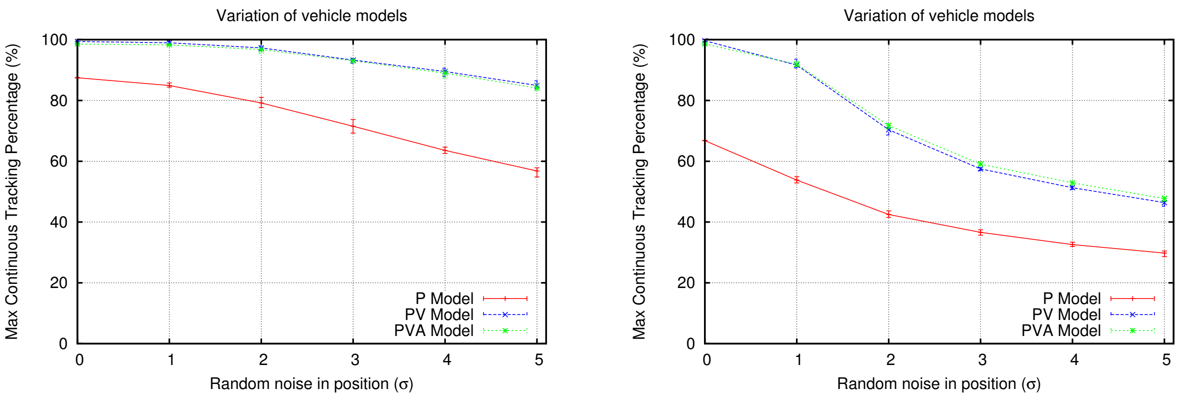
As a comparison with the results shown in [8], their tracker accuracy is degraded so much (up to 40%) for any random noise and for beacon intervals more than one second even with small densities (75 vehicles and higher). However, our achieved accuracy is still over 60% for noises of standard deviations up to two meters and over 70% with beacon time intervals up to three seconds for all evaluated densities. These differences can arise from the tracking method, the simulation scenarios and configurations and the vehicle state model. First, it is unlikely to have an accuracy from the NNPDA better than that from the MHT. The MHT tries multiple hypotheses over subsequent time steps rather than taking an assignment decision based on the information of the current time step as in the NNPDA. Thus, the tracking method is not the essential reason for the accuracy degradation. Regarding the simulation scenarios, they used scenarios generated from STRAW vehicular mobility model which is different from the driver behavior model used in VISSIM simulator. However, we notice that the accuracy of our tracker in the highest dense scenario we used is still more than theirs in the lowest dense scenario they use with the similar position noise and beacon interval. As it is unlikely to have a more challenging case in the sparsest scenario than in the densest scenario, scenario differences are not the accuracy degradation reason. This means the vehicle model used in tracking may be the reason. We assumed that the measurement in the vehicle model is based on the position, velocity and acceleration included in beacons, while they used the position only.




(a) Highway scenario

(b) Urban scenario

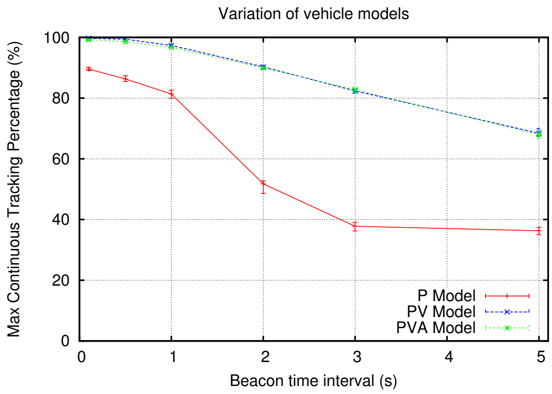
Figure 13: Vehicle entrance rate versus variations of vehicle models



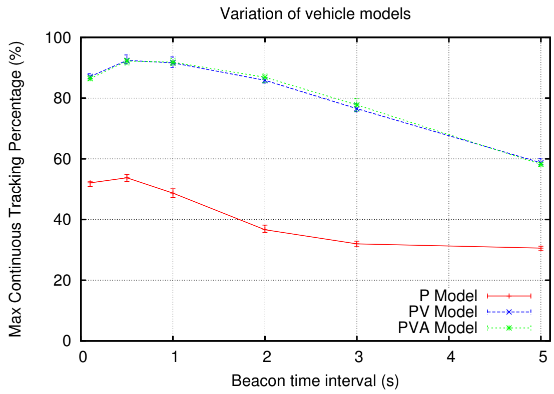

(a) Highway scenario

(b) Urban scenario

Figure 14: Random noise in position versus variations of vehicle models




(a) Highway scenario




(b) Urban scenario

Figure 15: Beacon time interval versus variations of vehicle models

To evaluate the model effect, we modified our vehicle model twice to use the position information only (P Model) and to use the position and velocity (PV Model). Thus, the P Model is defined as:

$$x_k = \begin{bmatrix} p \\ v \end{bmatrix}, A = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}, z_k = \begin{bmatrix} p \end{bmatrix}, H = \begin{bmatrix} 1 & 0 \end{bmatrix}, Q = \begin{bmatrix} t^4/4 & t^3/2 \\ t^3/2 & t^2 \end{bmatrix} \sigma_v^4, R = \begin{bmatrix} \sigma_p^2 \end{bmatrix} \tag{13}$$

While the PV Model is defined as:

$$x_k = \begin{bmatrix} p \\ v \\ a \end{bmatrix}, A = \begin{bmatrix} 1 & t & t^2/2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix}, z_k = \begin{bmatrix} p \\ v \end{bmatrix}, H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{14}$$

$$Q = \begin{bmatrix} t^4/4 & t^3/2 & t^2/2 \\ t^3/2 & t^2 & t \\ t^2/2 & t & 1 \end{bmatrix} \sigma_{as}^4, R = \begin{bmatrix} \sigma_p^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix} \tag{15}$$

It is worthy to note that the previous matrices are for a single coordinate. Also, the acceleration is included in the state of PV model as it will lead to a better performance than if it is not included as claimed in [13].

We run the tracker using the modified models along with the original one (PVA Model) on the highway and urban scenarios with similar parameters specified in Tables 1 and 2. As shown in Figures 13, 14 and 15, the P model performs worse than the other models with different entrance rates, position noises and beacon time intervals. This confirms that the degradation in the tracking accuracy in [8] caused by the model they used. Thus, we can also conclude two important findings from this result. First, position information is not sufficient to achieve reliable vehicle tracking. Second, using position and velocity information is sufficient for vehicle tracking and gives similar accuracy as using acceleration additionally.

On the other hand, we think that the tracking accuracy presented in this paper can be further enhanced in several ways. First, beacons contain additional static data, such as vehicle type and size. If this information is additionally used, it will help in discriminating between vehicles when the tracker is confused as it is likely to have vehicles with different characteristics near each other in the same time. Beside static data, dynamic data that changes every period of time, such as pseudonym and communication addresses, can be also utilized during its non-changing intervals, if vehicles do not swap them.

Second, the position broadcast in beacons is assumed to be more accurate than the value received solely from GPS leading to more accurate tracking. The GPS value is commonly augmented in VANET using cooperative positioning such as DGPS, SBAS and even using V2V and V2I communications [22]. Moreover, vehicular positions is also matched to road maps for navigation purposes using *map matching* algorithms. A map matching algorithm integrates the position with spatial road networks to identify which road and lane the vehicle travels in [23]. These positioning enhancements assert a higher tracking accuracy.

Third, exploiting the road network can be extended to the tracking filter itself. Road map assisted ground target tracking is getting more concern in research [24, 25, 26]. As most of vehicles move on roads, it is a good choice to incorporate road maps into the tracking process. Road curvature and surface, velocity limit and road direction are suggested constraints on the estimated states gained from the tracking filter. These constraints lead to better estimations which in turn lead to better data association and tracking accuracy.

# 5 Conclusion

In this paper, we implemented and evaluated a vehicle tracker. Based on the shown results, we can conclude the following findings. First, the main factors affecting the vehicle tracking are the vehicle density and then the random noise in position. Although large random noises in sparse environment may lead to great tracking accuracy, common smaller noises ($\sigma \leq 2$ m) in denser networks decrease such accuracy a lot. Second, larger beacon time intervals decrease the tracking accuracy, but the range of required beacon intervals in safety applications (up to 2 seconds) has only a slight impact. Third, position and velocity information are the necessary and sufficient information to be able to effectively track vehicles using their beacons. Forth, the packet loss degrades the tracking accuracy greatly, however, such effect is reduced with the most frequent beacon rate (10 Hz). On the other hand, vehicles can still be perfectly tracked even in challenging environments with intermediate tracking accuracy. These findings along with further possible enhancements to the tracking show the essential need for securing the beacon messages from being globally public. Further investigation is required to find the effective and efficient way to make beacons secure and private.

# References


[1] K. Emara, W. Woerndl, and J. Schlichter, "Vehicle tracking using vehicular network beacons," in *Fourth International Workshop on Data Security and PrivAcy in wireless Networks 2013 (D-SPAN 2013)*, (Madrid, Spain), June 2013.

[2] B. Parno and A. Perrig, "Challenges in securing vehicular networks," pp. 1–6, Nov. 2005.

[3] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung, and J.-P. Hubaux, "Secure vehicular communication systems: design and architecture," *Communications Magazine, IEEE*, vol. 46, no. 11, pp. 100 –109, 2008.

[4] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *Journal of Computer Security*, vol. 15, no. 1, pp. 39–68, 2007.

[5] J. Freudiger, M. Raya, M. Flegyhzi, P. Papadimitratos, and J.-P. Hubaux, "Mix-Zones for Location Privacy in Vehicular Networks," in *ACM Workshop on Wireless Networking for Intelligent Transportation Systems (WiN-ITS)*, (Vancouver), Aug. 2007.

[6] D. Eckhoff, R. German, C. Sommer, F. Dressler, and T. Gansen, "Slotswap: strong and affordable location privacy in intelligent transportation systems," *Communications Magazine, IEEE*, vol. 49, pp. 126 –133, Nov. 2011.

[7] R. Lu, X. Li, T. Luan, X. Liang, and X. Shen, "Pseudonym changing at social spots: An effective strategy for location privacy in vanets," *Vehicular Technology, IEEE Transactions on*, vol. 61, pp. 86 –96, Jan. 2012.

[8] B. Wiedersheim, Z. Ma, F. Kargl, and P. Papadimitratos, "Privacy in inter-vehicular networks: Why simple pseudonym change is not enough," in *Wireless On-demand Network Systems and Services (WONS), 2010 Seventh International Conference on*, pp. 176 –183, Feb. 2010.

[9] P. Golle and K. Partridge, "On the anonymity of home/work location pairs," in *Proceedings of the 7th International Conference on Pervasive Computing*, Pervasive '09, (Berlin, Heidelberg), pp. 390–397, Springer-Verlag, May 2009.

[10] Q.-J. Kong, Q. Zhao, C. Wei, and Y. Liu, "Efficient traffic state estimation for large-scale urban road networks," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, pp. 398–407, Nov. 2013.

[11] J. Singh, "Technique for privacy preserving real-time vehicle tracking using 802.11p technology," in *Proceedings of the 9th International Conference on Advances in Mobile Computing and Multimedia*, MoMM '11, (New York, NY, USA), pp. 206–209, ACM, Jan. 2011.

[12] V. S. C. Consortium, *Vehicle Safety Communications Project: Task 3 Final Report: Identify Intelligent Vehicle Safety Applications Enabled by DSRC.* National Highway Traffic Safety Administration, Office of Research and Development, Washington, D.C., 2005.

[13] S. Blackman and R. Popoli, *Design and analysis of modern tracking systems.* Artech House radar library, Artech House, Aug. 1999.

[14] S. Yaakov Bar-Shalom, n. Peter K. Willett, and r. Xin Tian, *Tracking and Data Fusion: A Handbook of Algorithms.* YBS Publishing, Apr. 2011.

[15] R. Kalman *et al.*, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[16] G. Welch and G. Bishop, "An Introduction to the Kalman Filter: SIGGRAPH 2001 Course 8," in *Computer Graphics, Annual Conference on Computer Graphics & Interactive Techniques*, pp. 12–17, 2001.

[17] Y. Bar-Shalom, F. Daum, and J. Huang, “The probabilistic data association filter,” *Control Systems, IEEE*, vol. 29, pp. 82 –100, Dec. 2009.

[18] D. P. Bertsekas, “The auction algorithm for assignment and other network flow problems: A tutorial,” *Interfaces*, vol. 20, pp. 133–149, July 1990.

[19] R. J. Fitzgerald, “Development of practical pda logic for multitarget tracking by microprocessor,” in *American Control Conference, 1986*, pp. 889 –898, jun 1986.

[20] PTV, Karlsruhe, Germany, *PTV VISSIM 5.10 User Manual*, July 2008.

[21] R. Wiedemann, “Simulation des straenverkehrsflusses,” *Schriftenreihe des Instituts fr Verkehrswesen der Universitt Karlsruhe*, no. Heft 8, 1974.

[22] N. Alam, A. Tabatabaei Balaei, and A. Dempster, “Relative positioning enhancement in vanets: A tight integration approach,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, pp. 47–55, Mar. 2013.

[23] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk, “On map-matching vehicle tracking data,” in *Proceedings of the 31st international conference on Very large data bases*, VLDB ’05, pp. 853–864, VLDB Endowment, Aug. 2005.

[24] D. Streller, “Road map assisted ground target tracking,” in *Information Fusion, 2008 11th International Conference on*, pp. 1–7, July 2008.

[25] C. Yang, M. Bakich, and E. Blasch, “Nonlinear constrained tracking of targets on roads,” in *Information Fusion, 2005 8th International Conference on*, vol. 1, pp. 8 pp.–, July 2005.

[26] M. Ulmke and W. Koch, “Road-map assisted ground moving target tracking,” *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 42, pp. 1264–1274, Oct. 2006.