Improving Neighbor Localization in Vehicular Ad Hoc Networks to Avoid Overhead from Periodic Messages

Azzedine Boukerche, Cristiano Rezende, Richard W. Pazzi
PARADISE Research Laboratory - University of Ottawa - Canada
{creze019,rwerner,boukerch}@site.uottawa.ca

Abstract—Vehicular Ad Hoc Networks (VANETs) have been widely studied, and the deployment of such networks is likely to happen soon as the required technology and commercial opportunities are already available. Most of applications proposed for these networks require a localization mechanism with reasonable accuracy. In addition to applications, most protocols rely on the availability of a system that determines vehicles’ positions. As many vehicles are (or are in the process of being) equipped with GPS, the accurate localization of vehicles in VANETs is achievable.

However, one issue that has been ignored is the localization of neighboring vehicles. In VANETs, nodes usually move at high speeds; the information about the position of a neighbour vehicle is therefore fast outdated. The naïve approach for handling this issue is to increase the frequency by which periodic messages containing a node’s position are exchanged. This solution might lead an unfeasible overhead significantly dropping the bandwidth available for the exchange of services’ data.

In this paper, we study this problem further, and propose a solution where vehicles predict the position of a neighbor for the near future. Through extensive experiments, we show that a prediction model of low complexity was able to considerably increase the accuracy of neighbor localization. Our mechanism has achieved an accuracy of 50 centimeters with a frequency of exchange of beacons 75% smaller than the naïve approach.

I. INTRODUCTION

Dedicated Short Range Communications (DSRC) is a standard for short to medium communication between vehicles using the 5.9GHz frequency. This technology can be used for the deployment of the network architecture know as Vehicular Ad Hoc Networks (VANETs). DSRC radios can be used for the communication among vehicles, in what is known as vehicle-to-vehicle (v2v) communication; or the exchange of messages between a vehicle and a roadside infrastructure, name vehicle-to-roadside (v2r) communication. This work is based on the v2v communication architecture, however, with simple modifications it could easily be suitable to v2r communication as well.

VANETs behave distinctly when compared to other infrastructureless networks. The former posses more powerful hardware and it is not afflicted by energy restrictions, though, its topology is considerably more dynamic topology due to the high speed by which nodes move along the network. These peculiarities demand different approaches in order to achieve better results than the previously proposed solutions.

Several applications can be developed on top of these networks, and they are usually aimed at either enhancing safety, improving emergency response or providing general services for drivers/passengers. Examples of applications that increase the safety of drivers and passengers are emergency brakes activated when a nearby accident is detected, or an automatic way to assure a minimum distance between cars. Proper healthcare in an emergency response can be given faster if, for instance, a live video stream is provided to a paramedic inside of an ambulance on the way to the accident site. With DSRC, it is also possible to provide Internet access inside the vehicle or even videoconferencing between drivers or passengers in different vehicles.

The vast majority of applications and protocols designed for VANETs assume that a vehicle’s geographic localization is already provided. Nodes’ position is used through-out many layers of the network stack, from the MAC layer [1], through routing [2] to the applications previously mentioned. Therefore, for the deployment of these networks in a real environment, it is fundamental that all issues related to the localization of vehicles be understood and fully addressed.

There are many localization approaches that mainly handle the trade-off between cost (e.g. bandwidth, calibration, equipment price) and accuracy. In other networks with no infrastructure, the cost plays a more crucial role than in DSRC-based networks; thus, the later adopts mechanisms that provide more precise information. Killer applications, such as collision avoidance, driver assistance and automatic parking, require highly accurate localization schemes [3], [4], sometimes even more precise than that provided by standard GPS.

Despite this need for accurate location information, to the best of our knowledge, there is no work in the literature that adequately deals with the problem of outdated information regarding neighbors position in mobile networks. This issue is aggravated in these networks because the location of a neighbor changes very fast. Those existing protocols that require precise locations handle this issue by increasing the frequency of periodic messages (beacons) that in turn contain the nodes’ positions. However, this solution leads to a high number of messages exchanged, and, even worse, to a high channel occupancy and to an increased number of collisions.

Our attempt to handle this issue is based on including enough information into beacons for vehicles to not only know the sender’s current position, but also for them to predict its location in the near future. We understand that the movement...
of vehicles can be reasonably predicted for the near future (by a few seconds) based on information such as recent past movement, neighbors’ movement, roads restriction or popular routes. Through prediction, we intend to exchange periodic messages at a lower frequency, then occupying the channel for a much shorter time period while providing a more accurate localization of nodes’ neighbors.

The following section contains a review of existing localization solutions for VANETs and of works related to our protocol. In Section III, we describe in details our protocol, which is evaluated and discussed in Section IV. Finally, in Section V, we present our final remarks.

II. LOCALIZATION MECHANISMS IN VANETs

The most common and most widely deployed localization mechanism is the triangulation signals received from orbiting satellites. Although there are other proposed systems (e.g. Galileo, GLONASS), the one which is currently most used is the Global Positioning System (GPS) [5]. Many vehicles already come with the necessary equipment for GPS, and several others have been equipped as well. It provides a reasonable accuracy, with errors from 10 to 30 meters; however, this can be unsuitable for applications that require high precision. Nevertheless, the errors of GPS receivers close to each other are correlated, and this property is used by Differential GPS (DGPS) [6] in order to provide sub-meter accuracy.

Another well-studied approach to nodes localization in a wireless environment is through analysis of the communication between nodes and base stations. This is called “trilateration” or “multilateration” and it requires that the node’s radio is within reach of at least three base stations (for a 2D position). This approach is based on the concept that it is possible to calculate a node’s position if the distance or angle between this node and three non-collinear fixed points whose positions are known. The distance between a node and a base station can be inferred by the strength of the received signal (this technique is called Received Signal Strength - RSSI) or by the time a packet takes to travel from the node to the base station (technique called Time of Arrival - ToA). Besides that, the angle by which the signal arrived can be used well in a technique called Angle of Arrival - AoA. However, trilateration is not accurate enough for most applications designed for VANETs, with errors up to 250m [7], [8].

Although there are many studies on how to provide accurate localization for VANETs, there are no studies (to the best of our knowledge) regarding how to handle the problem of the position of nodes becoming quickly outdated when stored by neighbors. A naive solution to this problem would be to increase the frequency by which the periodic messages (beacons) containing nodes’ positions is broadcast. However, this would likely lead to an overhead of messages, specially in a dense scenario, causing many packets collisions and substantially decreasing the overall bandwidth. In [9], the authors suggest that, under a congested scenario, these periodic messages should be sent with a lower power in order to provide a fair sharing of bandwidth in the network. This solution only decreases the density of the network and partially handles the impact of excessive beacons but does not deal with outdated information at the nodes. In the following Section, we propose how to better handle this issue.

III. PROPOSED NEIGHBOR LOCALIZATION PROTOCOL

All previously mentioned localization schemes focus on nodes’ own position accuracy. However, in most cases the location of a vehicle in a VANET is used by neighboring nodes as well. Greedy location-based routing algorithms [10], for instance, require that each node know the position of every single neighbor. Moreover, most location-based routing algorithms require that at least the source node know the position of the destination node, which may be more than one hop away from it [11]. Therefore, it is crucial that vehicles are made aware not only of their own precise location, but also of the accurate position of other nodes.

In this work, we tackle the problem of how to make available accurate localization of one-hop neighbors at any moment for any vehicle. Previous solutions only increase the frequency by which localization related information is broadcast to neighbors and they are not suitable, since they are not scalable to the dense scenarios common in DSRC environments (due to traffic, accidents, stop lights, and so on). Furthermore, vehicles may rely on inaccurate positions when two previously reachable nodes move away from each other and can not update their mutual location information anymore.

Our solution consists of adding to beacons, besides the usual node’s position, information that can be used to predict a vehicle’s location into the near future. The furthest into future a vehicle would have to predict another vehicle’s position is until a node is removed from a node’s neighbors table. This time is usually within a few seconds. Additional information that can be stored in beacons are previous positions, speed and direction, or even routes within a shared road or city

\[1\] In order to extend the proposed protocol to solve the issue of keeping accurate localization of other nodes (e.g. a destination node), the only modification is to means of forwarding the proper packets in a multi-hop manner.
map. Global shared information may also be used in order to predict vehicles’ future positions, such as popular chosen roads, avenues and streets; traffic conditions; or maps that delimit vehicles’ movements.

Several distinct prediction models may be used for the purpose of estimating vehicles’ positions. They can be as simple as vector estimations based on previous positions and speed, or as complex as Markov Chains and Artificial Neural Networks. In order to increasing accuracy, prediction models tend to become more complex, requiring higher processing power, memory or bandwidth. The most appropriate model for any scenario is the one that better exploits the trade-off between accuracy and cost.

Algorithm 1 Broadcasting Beacon

1: Request Prediction Module for information required to predict this node’s position
2: Assemble Beacon with current position, information required for prediction, Beacon Sequence Number and any other periodic information that must be exchanged
3: Broadcast Beacon
4: Request Prediction Module to keep track of distance between this node’s predicted position and current position
5: Increment Beacon Sequence Number

Algorithm 2 Upon Received Beacon

1: Forward the Beacon to any other module that requires periodic information
2: Forward the Beacon to Neighbors Table
3: if Source Node is not on Neighbors Table then
4: Create new entry on Neighbors Table
5: Request Prediction Module to create new object to predict source node’s position based on the information within the Beacon
6: else
7: Update neighbor timestamp on Neighbors Table
8: Forward new information for prediction from Beacon to the Prediction Module
9: end if

In our protocol, whenever a vehicle requires the location of another vehicle, it will use the predicted position instead of the one initially informed through the beacon. Our goal is to improve the accuracy of the localization of neighboring vehicles while decreasing the necessary number of beacons.

Another mechanism we have implemented ensures that, when a node sends the beacon with its position and the information necessary for the other vehicles to predict its future positions, the sender also store this information, thus, determining the error between the predicted and the actual positions. Consequently, we can establish the threshold $\delta$, which is used to trigger the broadcast of a new beacon with updated information. In other words, when the difference between the predicted and actual positions is greater than $\delta$, the node assembles a new beacon with the current information and broadcasts it updating neighbors’ predictions. If global information is used for the prediction of vehicles’ positions, this mechanism can only be used if this information is shared among all nodes involved.

Figure 1 illustrates the flow of information and triggers inside a node which happens in two circumstances to broadcast a new beacon: periodic or due to large error between predicted and real position. Since we consider the localization mechanism as an essential aspect of VANETs, and the exchange of messages is solely among reachable nodes, this protocol can be incorporated into the MAC layer. Algorithms 1 and 2 describe the procedure to broadcast a beacon and the actions taken when one is received, respectively.

An example of how our proposed protocol works is shown on Figure 2. In the first step $t_0$, a beacon is broadcast by the mobile node. In this initial moment, our proposed solution based on prediction would estimate the same position as the naïve approach, both would consider the real position since the location information is updated. On the second step $t_1$, the prediction model at a neighbor node would use the information attached to the beacon to estimate the position of the node. Although the estimation $P_3$ is not perfect, it is certainly better than the naïve solution $P_1$. With time, the distance between the actual position and the estimated one tends to increase (Fig. 2(c)). In the last step $t_3$, another beacon is sent by the node, thus, the localization information is updated and the cycle starts again.

IV. Performance Evaluation

In order to evaluate our proposed protocol’s effectiveness in exploiting the trade-off between accuracy and cost for the localization of neighbors in a DSRC-based network, we have implemented using Network Simulator (ns2) [12]. In the following subsection, we describe our simulation environment in detail. In Section IV-B, we explain the statistical methods used and the metrics evaluated, and we also present the results of the simulation. Then, in Section IV-C, we discuss these results.

A. Simulation Setup

In Table I, we summarize the values of the parameters used in our experiments. Note that the radio power (which determines the radio range) is the default value and it is always used unless a different value is specified.

The prediction of future positions is strongly related to a vehicle’s movement; since we are evaluating our idea through simulations, it was important that we use a reliable and realistic mobility model. We have opted to evaluate our mechanism in a road scenario, since mobility models can represent this environment with higher fidelity. The most commonly used model for a highway scenario is the Freeway[13] model. However, an unrealistic aspect of this model is that vehicles do not cross other vehicles once they reach a slower vehicle in front of them; instead they reduce their speed once they are within a certain threshold distance from this vehicle. The
resulting scenario would be characterized by a synchronization of speeds and a constant distance between vehicles in the same lane – that is, more stable topologies. We have altered this model so that it supports faster vehicles crossing slower ones. The enhanced model, Freeway\textsuperscript{+}, is composed of straight lanes for both directions (curves do not need to be represented due to the short range communications inherent in these networks) and the same number of lanes per direction. Each lane has a minimum and maximum speed, and vehicles speeds are chosen randomly and change every \(s_c\) seconds. When nodes reach one of the two extremes, a reset method is called, and the nodes are replaced on the other extreme and in the same lane. Nodes are distributed uniformly on the road – that is, always within only one lane. Table I shows the parameters of the mobility scenario used in this work.

Many results from simulations revealed the impact of the Cold Start effect, where the scenario at the beginning of the simulation has still not reached a stable state. We have ensured three conditions in order to avoid this effect: i) in the first 10 minutes nodes only move around; ii) results are based on exchanges of messages that occurred after the first kilometer and before the last one, to avoid road extremities issues, thus, the evaluated road length is 10km with 50 nodes per kilometer and an average of 500 nodes; and iii) the first 10 minutes of beacons exchanges are performed but not evaluated. With these measures in place, we provide the results free of Cold Start and extremes bias. The 60 minutes of simulation considered are, therefore, the 60 minutes after the 20 initial used to avoid these aforementioned issues.

Our focus is on how to provide accurate localization of vehicles, from the perspective of a neighbor vehicle. We do not handle the localization issue of the node itself from its perspective. For this reason, we do not consider the error of the initial localization mechanism. In our scenario, vehicles are able to obtain their own position precisely. The initial vehicle localization is beyond the scope of this work.

The prediction model is also an important part of our simulation. We have decided to use a model with low complexity so that beacons would be of only a few bytes and would not incur much overhead, in terms of both bandwidth and computational power. The model here involves calculating a vector \(v = (s_x, s_y)\) that indicates movement direction and speed based on the last broadcast position \(p_0 = (x_0, y_0)\) at time \(t_0\) and the current one \(p_1 = (x_1, y_1)\) at time \(t_1\). Therefore, the vector \(v\) can be calculated as shown:

\[
s_x = \frac{(x_1 - x_0)}{(t_1 - t_0)} \quad \text{and} \quad s_y = \frac{(y_1 - y_0)}{(t_1 - t_0)}
\]

A beacon holds, in addition to the node id, the position \(p_1\) and the vector \(v\. Whenever a beacon is received, this information and the current timestamp are stored. When a vehicle \(n_r\), that has received the beacon from node \(n_s\) at \(T_0\), wants to use the geographic position \(n_s\) at time \(T_1\), the position \(P = (x_p, y_p)\) here used is given by the following equations:

\[
x_p = x_1 + s_x * (T_1 - T_0) \quad \text{and} \quad y_p = y_1 + s_y * (T_1 - T_0)
\]

It is important to notice that no global time synchronization is needed since the speed can be calculated by any node using only the time difference between consecutive beacons, and since the predicted position is based only on the difference between the current local time and the local time when the beacon was received. This can be done because the delay of a one-hop transmission is inexpressive when compared to the movement of a vehicle.

One parameter that fundamentally influences the performance of our protocol is the frequency \(f\) by which beacons are sent. We have evaluated different frequencies by varying

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radio Propagation Model</td>
<td>TwoRayGround</td>
</tr>
<tr>
<td>Mac Layer</td>
<td>IEEE 802.11</td>
</tr>
<tr>
<td>RXThreshold</td>
<td>1.92278e-08(100m)</td>
</tr>
<tr>
<td>Antenna</td>
<td>OmniAntenna</td>
</tr>
<tr>
<td>Simulation Time</td>
<td>60 min</td>
</tr>
<tr>
<td># Nodes</td>
<td>600</td>
</tr>
<tr>
<td>Length</td>
<td>12km</td>
</tr>
<tr>
<td>Lanes per Direction</td>
<td>3 - 5m wide each</td>
</tr>
<tr>
<td>Speed Limits (m/s)</td>
<td>5-15,10P-25,20-40</td>
</tr>
<tr>
<td>(s_c)</td>
<td>60s ± 10%</td>
</tr>
<tr>
<td>(\delta)</td>
<td>0.5; 1; 3; 5; 10; 15 (s)</td>
</tr>
<tr>
<td>(\Pi; \Delta)</td>
<td>0.25,1; 0.5,1.5; 1.3; 5.15 (s)</td>
</tr>
</tbody>
</table>

TABLE I

SIMULATION PARAMETERS

![Fig. 2. This figure shows the movement of one node and the position considered by a neighbor node reachable for at least from \(t_0\) to \(t_3\). The table on the top left shows the real current position of the node (R), the predicted (P) and the one that would be used in a nave approach (N). A beacon is broadcast by the moving node at \(t_0\) and \(t_3\). It is clear that the predicted position is more accurate then the nave solution during the time between consecutive beacons are received.](image)
the beacon period $\Pi = 1/f$. Another important parameter is the time $\Delta$ that a vehicle remains on the another vehicle’s neighbors table if no other beacon is received during this time. The value chosen for $\Delta$ is strongly related to $\Pi$ and $\Delta$ is usually around three times $\Pi$. Table I shows the values used in this work for the distance threshold $\delta$ (see Section III) and the pair $\Pi, \Delta$. Besides these parameters, we have also evaluated two different values for the radio range $r$ (50m and 100m) in order to understand the impact of network density on the overall performance. Therefore, an instance we had run assumed a configuration that can be defined by the quadruple $(\delta, \Pi, \Delta, r)$ and there are 48 distinct possible configurations$^2$.

B. Experimental Results

Each configuration was executed 32 times; for this reason, we could use the Normal Distribution$^2$ to calculate the confidence intervals. Each point plotted on the following figures was an average of these 32 runs and, with all of them, confidence intervals at a confidence level of 99% are plotted.

We mainly want to evaluate the trade-off between accuracy and cost; accordingly, we have defined two metrics, one related to mean error and number of beacons, respectively. The error on any node is the average between all neighbors on the predicted position and the actual one. The mean error is the average of this error between all vehicles in the network. The number of beacons is the average number of beacons sent among all nodes in the network; therefore, it is a value per node and node for the whole network.

Our first analysis is of the performance of our protocol through time. In Figures 3(a) and 3(b), we notice that the behavior of the network, in terms of the two aforementioned metrics, does not alter through time. For this reason, in the remainder of this paper, the mean errors considered are the average of the 37 samples collected (one for each 100 seconds of the 1 hour simulation time plus one at the beginning). The number of beacons sent is the total number of beacons sent for the whole hour simulated.

Before we start to evaluate the performance of our protocol, it is important to point out that beacons are not only used to update localization information to neighbors, but also to update the reachability status among nodes. As mentioned, after a period of time (in this work, $\Delta$) during which a node does not receive any new beacon, it is removed from the node’s neighbors table since it is likely out of reach. Figure 3(c) shows the average number of neighbors in nodes’ neighbors table for different $\Pi$. The network density is of 50 nodes per km (see Table I), and as each node covers the length of $2r$ meters, the expected number of neighbors is 10 when $r = 100$ and 5 when $r = 50$. In Figure 3(c), we can see that as shorter as the period between two consecutive beacons is, the most reliable a node’s neighbors table in representing the real environment is. However, we here propose an accurate mechanism for neighbor localization, thus, the vehicles’ positions can be used to estimate whether two nodes are still mutually reachable.

Figure 3(d) shows the performance of our protocol regarding the cost. Each curve represents a configuration and it is labelled by the pair $(\Pi, r)$. The results follow the expected behavior where increasing $\delta$ the number of beacons decreases (converging to the scenario where beacons are exchanged only when $\Pi$ expires) and for shorter beacon periods $\Pi$, more beacons are sent. The radio range $r$ clearly has an impact on the number of beacons sent.

An interesting observation on the performance of our protocol comes from the analysis of the results plotted on Figure 3(e). This figure shows the mean error when the Error Threshold $\delta$ varies. For small values, when $\delta$ grows, the mean error increases too, as expected. However, for long beacon periods $\Pi$, if we continue to increase $\delta$, the mean error surprisingly begins to decrease. In order to understand the reason for this, it should first and foremost be clear that the neighborhood of vehicles in a road scenario is mainly composed of vehicles travelling in the same direction and with similar speeds. Moreover, prediction errors in our work in a road scenario happen mostly due to vehicles changing speed. Therefore, when a vehicle changes its speed, it changes its neighborhood as well. With small values of $\delta$, this change is partially informed to the vehicle’s current neighborhood, but the vehicle will soon not be able to properly inform its new movement pattern. With bigger values for $\delta$, the vehicles takes longer to inform its new speed and it broadcasts more accurate information before leaving its previous neighborhood. For this reason, for the remainder of this paper, all the results plotted have $\delta = 25m$.

In Figure 3(f), we compare the accuracy of our protocol with the naïve approach based solely on the positions informed by beacons. It is clear that adding a mechanism to predict vehicles’ positions greatly increases the accuracy of the localization of neighbors. The radio range $r$ has a greater impact on the naïve approach than on ours. The increase of error with longer periods between beacons is also slower when prediction is used than when it is not.

C. Discussion

In our experiments, we have observed the counter-intuitive result that a larger value for $\delta$ led to more accurate localization (see Section IV-B for a more detailed explanation). However, we believe that, in a scenario where vehicles move more erratically than in a road scenario (e.g. in an urban scenario), smaller values of $\delta$ would incur a smaller error than for large values. This unexpected behavior happens only because of vehicle proximity in road scenarios, as explained earlier.

The results have shown that position prediction dramatically increases the accuracy of the localization of neighbors. For the same period $\Pi$ between consecutive beacons, the mean error was at least half when prediction was used (for $\Pi = 5s$ and $r = 100m$, the error was less than one fifth of the naïve approach). For a mean error of only 50 centimeters, using prediction, a quarter of beacons are sent. More importantly,
we can achieve such accuracy in a feasible scenario where the wireless channel is not constantly occupied for beacons exchange as in the naïve approach. The results in this work were under a prediction model of low complexity. It is still possible to design other models that can achieve even more accurate results.

Nevertheless, beacons are not only used for localization purpose. An important purpose of beacons exchange in DSRC-based networks is to update vehicles’ neighbors table, thus, assuring that routing protocols do not rely on nodes that are no longer available. Therefore, although accurate localization of neighbors could still be achieved while decreasing the beacons broadcast frequency (using prediction), this decrease could result in an unreliable neighbors table. This issue is easily solved by using the vehicle’s position to determine the distance between a vehicle and its neighbors, and to use that information to remove distant nodes from the neighbors table.

V. CONCLUSION

We have tackled a serious issue present in Vehicular Ad Hoc Networks that has been neglected by most researchers. This problem is that of how to provide accurate vehicles localization from the perspective of a neighboring node. This is an important issue, since, for most applications and location-based routing protocols, the position of neighbor nodes is used as frequently as the own node’s position, if not more frequently.

Our solution to handle this issue is to add, into periodic messages that contain vehicle positions, information that can be used to predict nodes’ near future positions (a few seconds). By doing so, any time a vehicle requires the position of another vehicle within range, the position utilized is not that which was initially informed but instead a predicted position.

Through extensive simulations, we have shown that using a prediction model of low complexity was enough to achieve significantly more accurate positions. For the same frequency of exchange of beacons, we achieved mean errors that were at least 50% to 80% lower. For an accuracy of 50 centimeters from the locally obtained position, we could decrease the frequency of beacons from an unsustainable 4 beacons per second to a feasible one of 1 beacon per second.

REFERENCES