Light-Weight Laser-Aided Position Verification for CASCADE

Khaled Ibrahim, Michele C. Weigle and Gongjun Yan
Department of Computer Science, Old Dominion University
Norfolk, VA 23529–0162
{*ibrah_k, mweigle, ygongjun*}*@cs.odu.edu*

***Abstract*—In this paper, we present a light-weight position verification technique for VANETs. Our technique supplements the received signal strength (RSSI) technique with a laser rangefinder to reduce the inaccuracy incurred in the RSSI. We used our new position verification technique to tighten the security in CASCADE (Cluster-based Accurate Syntactic Compression of Aggregated Data in VANETs) data aggregation technique. The proposed technique can detect any malicious vehicle sending false position information in less than 30 seconds and detects the GPS equipment malfunction in less than one minute with very few false positives, even under a low penetration rate. The incurred overhead due to using the position verification technique with CASCADE has been shown to be negligible.**

I. Introduction

Advance notification of traffic congestion through vehicular communication has been proposed to allow drivers to avoid costly highway congestion. A Vehicular Ad-hoc Network (VANET) consists of nearby vehicles exchanging information with each other via wireless broadcast. Using the wireless channel efficiently is a challenging problem. Most VANET messages (*e.g.*, speed and location updates) are periodically broadcast by each vehicle. To disseminate information widely, the messages must be forwarded to vehicles outside the original sender's broadcast range. The more vehicles participating in the VANET, the larger the number of messages sent and the higher the probability of wireless collisions.

The goal of CASCADE (Cluster-based Syntactic Compression of Aggregated Data in VANETs) [1], [2] is to allow a vehicle to obtain an accurate view of upcoming traffic conditions. Vehicles pass information about traffic conditions ahead to vehicles behind them so that they will have timely notification of upcoming traffic conditions. Although CASCADE was shown to use the wireless channel efficiently, security aspects of the system were not fully addressed. Recent work in VANET security has introduced the idea of using certified public keys (*i.e.*, PKI) stored in a tamper-proof device [3] or running on a tamper-proof service [4] to authenticate vehicles. While these features keep an attacker from obtaining false identities, they do not prevent a malicious user from tampering with other parts of the system, which provide the data that will be digitally signed by components in the tamper-proof device or service. We focus on additions to CASCADE that can detect and defend against false data whether malicious or the result of equipment malfunction. We analyze the possible sources for sending false position information attack, and we group them into four categories. We propose a technique for verifying vehicles' claimed locations by supplementing the received signal strength (RSSI) technique with a scanning laser rangefinder. We show that our defense technique can quickly detect false inputs given to CASCADE and will quickly quarantine those vehicles providing the false input. In contrast to other position verification techniques that add a noticeable overhead, our technique adds a negligible overhead to CASCADE performance, measured as MAC delay, reception rate and view update delay.

II. Related Work

Many studies have been performed for securing vehicular networks using either Public Key Infrastructure (PKI) [3], [5]–[8] or digital signatures [3], [9]–[11]. But none of these studies address how to verify the correctness of the data inside the messages, which is the main focus of our presented technique in this paper.

The position verification problem is one of the well-studied problems in the mobile ad-hoc networks (MANETs) [12], [13]. As VANETs have some very different characteristics from MANETs, namely higher mobility speeds and restricted network topologies and vehicle movement, the proposed solutions for the position verification problem in MANETs are not applicable for VANETs.

Leinmüller *et al.* [14], [15] showed that disseminating false position information in VANETs has a much more severe impact on the system in highway scenarios than in city scenarios. Thus, we focus on position information in highway scenarios.

There has been much recent work on position verification in mobile ad-hoc networks, as well as vehicular networks. Sastry *et al.* [12] proposed a technique to determine if a mobile communicating node exists in the region near where it claims to be. The limitation of this technique is that it cannot verify that the node is in the exact claimed location, which reduces its accuracy. Suen *et al.* [13] used radio signal properties (*e.g.* direction, strength) to determine the transmitter location. Using the radio signal properties alone bounds the technique's accuracy to the RSSI accuracy, which is in meters. In the previous two techniques, accuracy was an issue. Having such low accuracy will increase the false positive rate while our proposed technique inherits the laser rangefinder accuracy, which is in centimeters.

Xiao *et al.* [16] also use radio signal strength to verify nodes' locations. To alleviate the inaccuracy in this technique, they enhance the location estimation by using statistical algorithms and road-side infrastructure. The use of road-side infrastructure

increases the deployment cost drastically, while our proposed technique needs no infrastructure and has comparable accuracy.

Golle *et al.* [17] use heuristics such as drastic changes in speed or location to determine if a vehicle is sending false information. The technique tries to correct the information rather than quarantining the malicious node. In addition, the technique assumes that nodes can be distinguished from other nodes, using RSSI, camera, or other devices.

Yan *et al.* [18] proposed a novel technique for verifying the claimed location using an omni-directional radar system, but such a device is not yet available. Their technique incurs more communications overhead than our proposed technique, and some attacks cannot be handled satisfactorily (*e.g.*, a lying vehicle claims to be in a location where another vehicle already exists, so both are blocked).

III. Overview of CASCADE

A. Primary Frames

Every 300-400 ms, each vehicle broadcasts a *primary frame* containing its position information (including location, speed, acceleration, and heading). Receiving vehicles store this information in a *primary record*. Those primary records representing vehicles a certain distance ahead comprise the receiving vehicle's *local view*. The length of the local view is determined by the parameters set for aggregation, as all data must fit into a single frame (2312 bytes for IEEE 802.11). If the length of the local view is greater than the transmission range (assumed to be about 300 m), the primary frames are re-broadcast using probabilistic-IVG (p-IVG) [19], an enhancement of Inter-Vehicle Geocast (IVG) [20] that considers the density of surrounding vehicles in determining which vehicle will re-broadcast the frame.

B. Aggregated Frames

To distribute its local view to vehicles behind, every four seconds each vehicle will compress, aggregate, and broadcast the primary records in its local view as an *aggregated frame*. To facilitate compression and aggregation, each vehicle divides its local view into fixed-sized clusters, positioned relative to the aggregating vehicle. In CASCADE, the optimal cluster size was determined to be 16 m x 126 m, assuming a 4 m lane width [21].

Each time a primary frame is received, the described vehicle is grouped into its appropriate cluster, based on its heading and distance from the receiving vehicle. When the time comes for compression and aggregation, the location and speed of each vehicle relative to its cluster is represented in a *compact data record*. The position of each vehicle is translated into $\{X, Y\}$ coordinates (in integer meters) with the local view origin as point $(0, 0)$. Each compact data record contains the following fields:

- ΔX - difference between the vehicle's X coordinate position and the X coordinate for the center of its cluster in sign-magnitude representation
- ΔY -the difference between the vehicle's Y coordinate position and the Y coordinate for the center of its cluster in sign-magnitude representation
- ΔS (5 bits) - the difference between vehicle's speed and the median speed of the vehicles in the cluster in sign-magnitude representation
- *Speed Indicator (SI) Flag* (2 bits) - indicates if the vehicle's speed is within the acceptable range for the cluster or not ([-15 m/s, 15 m/s])

The range of acceptable values for ΔS is [-15 m/s, 15 m/s]. If the difference is outside of this range, then the ΔS field will be omitted, and the *SI Flag* will be set. The *SI Flag* can take one of three possible values $\{$`00`, `01`, `10`$\}$:

- `00` - ΔS can be represented in the allowed range [min ΔS, max ΔS]
- `01` - $\Delta S >$ max ΔS, the vehicle is a *speeder*
- `10` - $\Delta S <$ min ΔS, the vehicle is a *lagger*

Once the compact data records for all vehicles in the local view have been created, an *aggregated cluster record* is formed for each cluster, which contains the following fields:

- *cluster flag* (1 bit) - indicates if the cluster contains any vehicles
- *cluster median speed* (8 bits) - the median speed of the vehicles in the cluster in meters/second
- *number of vehicles* - the number of vehicles in the cluster
- *compact records* - concatenation of the compact data records in the cluster

If a cluster contains no vehicles, it is represented by a single bit (the *cluster flag* set to 0). Since the number of vehicles in a cluster depends upon the cluster size, we will discuss the optimal sizes for the last two fields later.

Once the aggregated cluster records are constructed, they are concatenated into a single frame and sent via broadcast. The *aggregated frame* contains the following fields:

- *type* (1 bit) - primary or aggregated frame
- *timestamp* (8 bytes)
- *sender's location* (16 bytes) - latitude/longitude
- *aggregating vehicle's X-coordinate* - meters from the vehicle's local view origin
- *aggregating vehicle's location* (16 bytes)
- *aggregated cluster records* - concatenation of all aggregated cluster records in the local view, arranged according to their position in the local view starting with the bottom-left cluster
- *digital signature* (28 bytes)
- *certificate* (56 bytes)

As a vehicle's primary data (location, speed) is represented in 136 bits and its compact data record is represented in at most 19 bits, CASCADE delivers a compression ratio of at least 86%.

IV. Contributions

In this paper, we analyze possible false position information attacks and categorize them according to the data source of the attack. Also, we present a new position verification technique that supplements RSSI with a laser rangefinder to reduce the inaccuracy incurred in RSSI, which usually results in a high false positive rate. Without loss of generality, we applied the proposed technique to CASCADE in order to evaluate the technique's performance and measure its incurred overhead

and, at the same time, to tighten CASCADE's security. The proposed technique proved to be fast in detecting any malicious vehicle in less than 30 seconds and detecting the GPS equipment malfunction in less than one minute even under low penetration rates. Also, it has a very low false positive rate making it highly accurate. Moreover, it is considered practical because the implementation cost is relatively low. Additionally, the incurred overhead resulting from applying the proposed technique to CASCADE under different vehicular densities was negligible

V. Position verification in CASCADE

With the goal of improving security in CASCADE, we present our technique for verifying the locations provided by vehicles about themselves in primary frames. The position verification technique takes advantage of recent advances in the use of scanning laser rangefinders for vehicular safety applications. In this section, we describe our threat model, how false data is detected, and how misbehaving vehicles are quarantined to prevent false data from spreading.

A. Assumptions

We assume that each vehicle is equipped with a GPS receiver using DGPS [22] with an accuracy on the order of centimeters, a communications device using DSRC [23], a scanning laser rangefinder, and the CASCADE application. Scanning laser rangefinders, such as those produced by SICK [24] and ibeo [25], have been recently used for vehicular safety applications [26], [27] to detect and measure the distance to objects with high accuracy. The ibeo LUX, for instance, can complete 25 scans a second with a range of 3-200 m and a field of view (FOV) of 100°. To cover 360° of FOV, the laser rangefinder may be installed on a rotating base. If this adds too much delay, multiple (up to 4) laser rangefinders could be installed. This is feasible, especially as the cost of a single laser rangefinder is only about 100 EURO. Each vehicle is also pre-assigned a public/private key pair and the public key's certificate, used for authentication.

In addition to using the scanning laser rangefinder to verify position, we also will use RSSI (Radio-based Signal Strength Indicator) provided by IEEE 802.11. With RSSI, a receiver can estimate the transmitter location using known mathematical models for the wireless channel. The estimated locations using RSSI are not highly accurate since there is a large variation in signal attenuation in different environments [28]. Despite some inaccuracies in location estimation, RSSI is widely used due to its low cost as it requires no special hardware [29], [30]. But, because of its inaccuracies, we use RSSI as only one part of our position verification technique.

We assume that since the system equipment (GPS, transceiver, and laser rangefinder) can be located anywhere in the vehicle, they are not tamper-proof. The only components that are tamper-proof are the CASCADE application and the new detection and quarantine modules, residing inside the tamper-proof device. Because of this, we only consider verifying the position of vehicles provided in primary frames. The aggregation component of CASCADE will be located inside the tamper-proof device and thus can be trusted.

B. Threat Model

Here we describe the types of attacks that could be perpetrated against the system. Since we assume that the CASCADE application itself is secure inside a tamper-proof device, attacks can only come from tampering with outgoing frames, tampering with received frames, or providing false input.

Tampering with outgoing frames will be easily detected because the vehicle's private key is not available outside of the tamper-proof device. Any change to the contents of outgoing messages will be detected when the receiver checks the digital signature on the message. This also eliminates the Sybil attack, where one vehicle impersonates multiple non-existent vehicles.

The only way a malicious user could tamper with received frames without detection is by dropping the frame before sending it to the CASCADE application. We will address this issue when we discuss how malicious vehicles are quarantined (Section V-C2).

The remaining attacks come from providing false position information to the CASCADE application. We classify these attacks into four categories according to the source of the false information:

- *Unsynchronized Trace:* The attacker could record a trace of position information during a previous trip on the same road and then substitute these recorded positions for the vehicle's actual current GPS coordinates. An unsynchronized trace is one where the positions being sent to CASCADE are outside the nominal transmission range from a potential receiver.
- *Synchronized Trace:* This attack is similar to the unsynchronized trace, but the claimed position is within the nominal transmission range of a potential receiver.
- *Primary Frame Replay:* The attacker could record the position information from recently received primary frames from other vehicles and then substitute these positions for its actual GPS coordinates. The difference between this attack and the previous two is that the attacker will be sure of having a vehicle in the claimed position which will make detection more difficult.
- *Malfunctioning GPS:* If the GPS device is malfunctioning, it will not be able to translate the received signal into the correct coordinates and will report incorrect location coordinates to CASCADE. Actually, this is not an intentional attack, but it has the same effect.

C. Defense Technique

Here we present how CASCADE will defend itself against the attacks described above. The proposed defense system consists of two main modules, the *detection module* and the *quarantine module*. The detection module is responsible for checking the consistency of received primary frames. If the detection module determines that the information in a primary frame is false, it will place the sending vehicle in the *suspect list* and trigger the quarantine module. The quarantine module in a vehicle detecting an attack is responsible for issuing a quarantine proposal and a quarantine request. Upon receiving a certain number of quarantine requests, the quarantine module

in a suspected vehicle is responsible for temporarily suspending the CASCADE application from sending any frames.

1) Detection Module: The detection module verifies that the location claimed in a received primary frame is consistent with the estimated location of the sending vehicle. Note that re-broadcasted primary frames (where the sender is not the original source) bypass this consistency check. The consistency check is performed in two levels: first using the signal strength (RSSI), and then using the laser rangefinder, if needed. The process of the detection module is described in Algorithm 1.

In the first-level consistency check, the vehicle's claimed location $rcvd\_loc$ is extracted from the received primary frame, and the location of the sending vehicle $rssi\_loc$ is estimated using RSSI. Also, the distance from the receiver to $rcvd\_loc$, $rcvd\_dist$, is compared with the nominal transmission range $xmit\_rng$. The next steps depend upon how the difference *rssi_diff*, which is the difference between $rcvd\_loc$ and $rssi\_loc$, compares to the RSSI accuracy threshold $rssi\_thr$[1]:

- *rssi_diff* $< rssi\_thr$
  The two locations are considered to be consistent and the received primary frame is added to the local view and processed normally by CASCADE.
- *rssi_diff* $> rssi\_thr \;\; \&\& \;\; rcvd\_dist < xmit\_rng$
  The vehicle's ID is placed in the suspect list, and the second-level consistency check is triggered.
- *rssi_diff* $> rssi\_thr \;\; \&\& \;\; rcvd\_dist > xmit\_rng$
  The second-level consistency check is bypassed, the vehicle's ID is placed in the suspect list, and the quarantine module is triggered.

In most cases, primary frames failing the first-level consistency check will be detected by multiple vehicles - all those that received the primary frame, as well. Each of these vehicles will check to see if the suspected vehicle claims to be a direct neighbor[2], using $rcvd\_loc$ and their local view. If the suspected vehicle is a direct neighbor, the detecting vehicle will begin the second-level consistency check, otherwise it will wait for notification from a direct neighbor of the suspect.

In the second-level consistency check, a direct neighboring vehicle of the suspect will use the readings from its scanning laser rangefinder to determine if there is a vehicle near $rcvd\_loc$. If there is a vehicle present, the location will be measured by the laser rangefinder as $laser\_loc$. The local view is checked to see if there is a different vehicle that claims to be at $rcvd\_loc$. If so, the vehicle is marked as *conflict*, otherwise it is marked as *unique*. As with the first-level check, the next actions depend upon how the difference, *laser_diff*, between $rcvd\_loc$ and $laser\_loc$ compares to the laser threshold $laser\_thr$[3]:

- *laser_diff* $< laser\_thr$ *&& unique*
  The vehicle is removed from the suspect list, added to the local view, and the frame is processed normally by CASCADE.
- *laser_diff* $< laser\_thr$ *&& conflict*
  The quarantine module is triggered.
- *laser_diff* $> laser\_thr$
  The quarantine module is triggered.

We note that a malicious user could falsify the information coming from the laser rangefinder. The false information could be used to either (1) deny the existence of an actual vehicle, or (2) confirm the existence of a false vehicle. To address the first case, we require that a majority of the neighboring vehicles be involved in quarantining a suspected vehicle; a single vehicle cannot achieve this. In the second case, this vehicle will not send a quarantine proposal, but other direct neighboring vehicles likely will, and the vehicle will still be detected.

Algorithm 1 Detection Module

```
Require: receiving a primary frame(pf)
 1: if pf.sender_loc = pf.originator_loc then {first level consistency check}
 2:     rcvd_loc ← extract_rcvd_loc(pf);
 3:     rssi_loc ← estimate_rssi_loc(pf.rcvd_signal_power);
 4:     rssi_diff ← |rssi_loc − rcvd_loc|;
 5:     rcvd_dist ← |rcvd_loc − current_loc|;
 6:     if rssi_diff > rssi_thr then
 7:         add pf.veh_id to suspect list;
 8:         if rcvd_dist > xmit_rng then
 9:             call Quarantine_Module;
10:         else
11:             isdirect_neighbor ← chk_localview(rcvd_loc);
12:             if isdirect_neighbor then {second level consistency check}
13:                 isdetected ← laser_detector(rssi_loc);
14:                 if isdetcted then
15:                     laser_loc ← estimate_laser_loc(rssi_loc);
16:                     laser_diff ← |laser_loc − rcvd_loc|;
17:                     isclaimed ← chk_localview(rssi_loc);
18:                     if laser_diff < laser_thr then
19:                         if isclaimed then
20:                             call Quarantine_Module; {conflict}
21:                         else {innocent}
22:                             Forward pf to CASCADE; {unique}
23:                         end if
24:                     else
25:                         call Quarantine_Module;
26:                     end if
27:                 else
28:                     call Quarantine_Module;
29:                 end if
30:             else {not direct neighbor}
31:                 do nothing;
32:             end if
33:         end if
34:     else
35:         Forward pf to CASCADE;
36:     end if
37: else
38:     Forward pf to CASCADE;
39: end if
```

2) The Quarantine Module: The operation of the quarantine module differs based on the current role of the vehicle. If a vehicle is a direct neighbor of a suspected vehicle, the quarantine module is responsible for sending quarantine proposals and quarantine requests. If a vehicle is a suspect, its quarantine module is responsible for suspending transmissions from the CASCADE application.

For direct neighboring vehicles, the quarantine module is triggered when a consistency check fails. When this occurs, the vehicle will broadcast a *quarantine proposal* (Figure 1).

[1]The value of the $rssi\_thr$ depends on the accuracy of the RSSI technique, in the range of meters.

[2]Vehicle *x* is considered a direct neighbor of vehicle *y*, if there is no vehicle between them, and *x* and *y* are within the laser rangefinder FOV of each other.

[3]The value of the $laser\_thr$ depends on the accuracy of the accuracy of the laser rangefinder, in the range of centimeters.

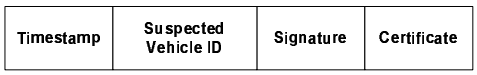

Fig. 1: The Quarantine Proposal Frame

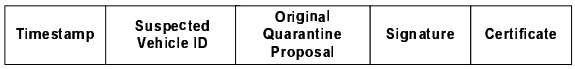

Fig. 2: The Quarantine Request Frame

The suspected vehicle may have more than one direct neighbor, which means that there may be more than one quarantine proposal sent at the same time. To avoid this, once a direct neighbor detects a lying vehicle, it will start a random countdown timer. If the timer expires and no other quarantine proposal about this suspect has been sent, the vehicle will send its proposal. This process is described in Algorithm 2.

The quarantine proposal is meant for vehicles other than the suspect. Upon receiving a quarantine proposal (Algorithm 3), a vehicle will check its suspect list for the vehicle ID contained in the proposal. If the vehicle is found in the suspect list, this vehicle will broadcast a *quarantine request* (Figure 2), meant for the suspect vehicle. Once the request has been sent, the suspect vehicle will be removed from the requesting vehicle's suspect list. This prevents multiple quarantine requests from being generated by the same vehicle upon receiving multiple quarantine proposals for the same suspect, which may only happen with low probability[4].

Upon receiving a quarantine request containing its ID (Algorithm 4) a suspected vehicle will initiate its pre-quarantine mode. In this mode, the suspect will initialize its quarantine request counter $Q\_req\_counter$ to 1 and count the number of vehicles N within its transmission range (this information is already contained in the vehicle's local view). Each time a new quarantine request is received, $Q\_req\_counter$ is incremented. The suspect vehicle moves from pre-quarantine mode to quarantine mode when $Q\_req\_counter > 0.5N$ and $Q\_req\_counter > 2$. This indicates that over 50% of the surrounding vehicles (and more than just two vehicles) have classified the vehicle as a suspect.

When the CASCADE application is started (which is assumed to be each time the vehicle is turned on), the number of times a vehicle has experienced a quarantine, Q_count, is set to 0. Each time quarantine mode is entered, Q_count is incremented. Once in quarantine mode, the suspected vehicle starts a quarantine timer that is set to expire in $60 * 2^{Q\_count-1}$ seconds (*i.e.*, exponential increase). The vehicle will be blocked from sending messages until the timer expires.

In order to avoid receiving quarantine requests, a malicious user might try to disable the receiving capability in its transceiver or discard all quarantine requests without sending them to the CASCADE application. Because CASCADE will never receive a quarantine request, the sending of frames will not be blocked, so bandwidth will be wasted with this vehicle sending potentially false primary frames. To counter this, CASCADE will periodically initiate a self-test to ensure that the transceiver is working properly. In the self-test, a dummy message will be sent out. This message consists of a quarantine request with a random nonce as the vehicle ID (not matching any vehicle in the local view). If a quarantine message with the same nonce is not received and delivered to the CASCADE application, the vehicle will enter quarantine mode immediately.

[4]This case may happen only if two direct neighbors of a suspect have their timers expire simultaneously. Given that the timer value is selected randomly, this will happen with low probability.

Algorithm 2 Quarantine Module

Require: receiving a primary frame (pf) from Detection Module
1: $timer \leftarrow random()$;{start a timer}
2: **while** $timer > 0$ **do**
3: wait for one second;
4: $timer - -$;
5: **end while**
6: $heard \leftarrow$ chk_rcvd_Q_proposal($pf.veh\_id$);{check if someone else send the same quarantine proposal}
7: **if** $not\ heard$ **then**
8: $Q\_proposal \leftarrow$ form_Q_proposal($pf.veh\_id$);{form a quarantine proposal}
9: send($Q\_proposal$);[send the quarantine proposal]
10: remove $pf.veh\_id$ from suspect list;
11: **end if**

Algorithm 3 Quarantine Module

Require: receiving a quarantine proposal (Q_proposal)
1: $found \leftarrow chk\_suspect\_list(Q\_proposal.veh\_id)$;
2: **if** $found$ **then**
3: $Q\_rqst \leftarrow$ form_Q_rqst($Q\_proposal.veh\_id$);{form a quarantine request}
4: send(Q_rqst);[send the quarantine request]
5: remove $pf.veh\_id$ from suspect list;
6: **else**
7: drop Q_proposal;
8: **end if**

VI. Evaluation

We evaluate our technique applied to CASCADE using ASH (Application-aware SWANS with Highway mobility) [31], which is an extension of the SWANS network simulator [32], [33]. SWANS fully implements the IEEE 802.11a protocol, which we use as an approximation to IEEE 802.11p. To produce realistic simulations, ASH includes implementations of the IDM (Intelligent Driver Model) vehicular mobility model [34] and the MOBIL lane changing model [35].

Table I summarizes our simulation setup. All vehicles have a transmission range of 300 m [36]. The roadway is a four-lane highway of length 100 km. Vehicles enter the highway according to a Poisson distribution and travel at a maximum speed of 30 m/s. The simulation is run for 360 seconds, resulting in a total of 500 vehicles generated. In the 360-second simulation runtime, the maximum distance traveled by any vehicle is 10 km.

We evaluate the position verification technique considering three different traffic density scenarios. High density traffic has an average of 90 vehicles/km, medium density has an average of 66 vehicles/km, and low density has an average of

Algorithm 4 Quarantine Module

Require: receiving a quarantine request (Q_rqst)
1: **if** $Q\_rqst.veh\_id = veh\_id$ **then** {start the pre-quarantine mode}
2: $timer \leftarrow$ random();{start a timer}
3: $density \leftarrow estimate\_denisty(trans\_range)$;{estimate the density within the transmission range}
4: $Q\_rqst\_counter \leftarrow 1$;
5: $ismalicious \leftarrow false$;
6: **while** $timer > 0$ and not $ismalicious$ **do**
7: wait for one second and store all Q_rqsts in Q_rqst_queue;
8: $timer - -$;
9: **while** Q_rqst_queue is not empty **do**
10: $new\_rqst\_rcvd \leftarrow$ chk_rcvd_Q_rqst($Q\_rqst.veh\_id$);{get a new quarantine request from the quarantine request queue}
11: **if** new_rqst_rcvd **then**
12: $Q\_rqst\_counter + +$;
13: **end if**
14: **end while**
15: **if** $\frac{Q\_rqst\_counter}{density} > 0.5$ **then**
16: $ismalicious \leftarrow true$;
17: **end if**
18: **end while**
19: **if** $ismalicious$ **then** {start the quarantine mode}
20: $Q\_count + +$
21: $Q\_timer \leftarrow 60 * 2^{Q\_count-1}$;
22: $block\_sending()$;
23: **while** $Q\_timer > 0$ **do**
24: wait for one second;
25: $Q\_timer - -$;
26: **end while**
27: $resume\_sending()$;
28: **end if**
29: **else**
30: drop Q_rqst;
31: **end if**

simulation runtime	360 seconds
highway length	100 km
vehicles generated	500
max speed	30 m/s
max distance traveled	10 km
transmission range	300 m
high density	90 vehicles/km
medium density	66 vehicles/km
low density	53 vehicles/km

TABLE I: Simulation Settings

53 vehicles/km. These densities were gathered from the speed and traffic volume analysis performed by Wisitpongphan *et al.* [37] for the data collected by the Berkeley Highway Lab for traffic on eastbound I-80 on June 27, 2006 [38].

As we mentioned in Section V-B, malicious vehicles can be grouped into four categories according to the source of the falsified information they send:

- *unsynchronized trace* - using old positions, but the claimed position is outside the transmission range of the receiver
- *synchronized trace* - using old positions, and the claimed location is inside the transmission range of the receiver
- *replay* - using other vehicles' position information as inputs to CASCADE
- *malfunctioning GPS* - using a malfunctioning GPS that will translate the GPS signals incorrectly

Unsynchronized trace	9 (2%)
Synchronized trace	15 (3%)
Replay	23 (5%)
GPS malfunction	12 (2%)
Total Malicious	59 (12%)

TABLE II: Number and Percentage of Malicious Vehicles in Each Category in the Simulation

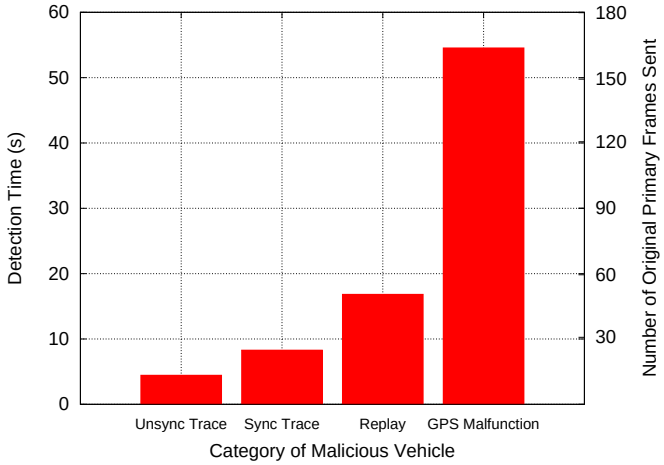

Fig. 3: Average Number of Original Primary Frames Sent and Average Time Elapsed Before First Quarantine for Each Category of Malicious Vehicle

The malicious vehicles are injected onto the highway during the simulation and represent 12% of the total vehicles in the simulation (Table II).

A. Defense with 100% Penetration Rate

The first experiment focuses on how well our proposed defense technique can detect malicious vehicles in each category, assuming all vehicles use CASCADE (100% penetration rate). In Figure 3, we show both the average number of original primary frames sent by malicious vehicles and the average lifetime before a malicious vehicle is detected and quarantined the first time. The number of messages sent represents the bandwidth wasted by malicious vehicles before being caught.

The *unsynchronized trace* attack is the easiest to detect because the vehicles are claiming to be in a location that is outside the nominal transmission range.

The *synchronized trace* attack is a little harder because the vehicles claim to be within the transmission range, so they will have to be detected using the laser rangefinder. If there is no vehicle in the claimed location, this attack will be easily detected. But, as the traffic density increases, the probability of having another vehicle in the claimed location also increases.

In the *replay* attack, a vehicle claims to be in another vehicle's location. This attack is similar to the previous case where another vehicle is actually in the claimed location, but now the probability of having another vehicle in the claimed location is 1. Two vehicles will claim the same location, but one vehicle will already be in the local view, while the other is not. In addition, the second vehicle will have a signal strength that does not exactly match its claimed location. Thus, the second

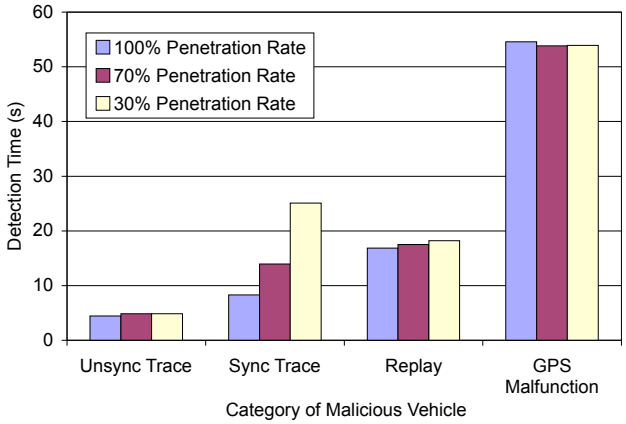

Fig. 4: Effect of the Penetration Rate on the Average Time Before the First Quarantine

vehicle (the malicious one) will be quarantined after costing the defense technique more time.

False information due to *GPS malfunction* takes the longest time to detect especially when the difference is neither large nor persistent.

B. Defense with Lower Penetration Rates

During the deployment of any VANET application, the penetration rate will be less than 100%. In this experiment, we explore the effect of having different penetration rates (70% and 30%) on the proposed defense technique.

In Figure 4, we show the detection time (*i.e.*, time until first quarantine) for each of the malicious vehicle categories. Having different penetration rates only greatly affects the detection time for the *synchronized trace* attack. As explained in the previous section, this type of attack is detected either because there is no vehicle at the claimed location or because a communicating, honest vehicle is already at that location and in the detecting vehicle's local view. With a lower penetration rate, the claimed location may be occupied by a non-communicating vehicle. In this case, the detecting vehicle will have no existing entry in its local view and will accept the vehicle as honest. The malicious vehicle will continue to go undetected as long as it can pick locations that are occupied by non-communicating vehicles. But, it is only a matter of time before the malicious vehicle is detected. In addition, there is little advantage to impersonating a non-communicating vehicle; the impersonator is only making it seem that the non-communicating vehicle is communicating and that the attacker is not communicating.

We show in Table III how the defense technique treats the honest vehicles in the system. About 5% of the honest vehicles are quarantined once, and only 2% are quarantined twice. Less than 0.3% of the honest vehicles are quarantined for a third time. Honest vehicles may be quarantined due to the inaccuracy of the RSSI, the detecting sensors, or a sudden change in the driver's behavior (*i.e.* fast lane change). Another way that an honest vehicle can be quarantined is through malicious attack. If one of the malicious vehicles is replaying the target vehicle's position and has been added to a detecting vehicle's local view, it will be seen as honest. When the honest vehicle appears in the area for the local view, it will be considered malicious (as there is already a supposedly honest vehicle in that location) and will be quarantined.

Quarantined 1 time	5%
Quarantined 2 times	2%
Quarantined 3 times	0.27%

TABLE III: Percentage of the False Positive Quarantines

C. Effect of Position Verification Overhead

To evaluate the effect of overhead added by the position verification technique on CASCADE performance, we measured MAC delay, frame reception rate, and view update rate with and without position verification.

1) MAC Delay: In order to investigate the impact of using the position verification (PV) with CASCADE on the wireless channel, we measured the amount of MAC-layer delay for each frame (split into primary frames and aggregated frames). This delay represents the time between the MAC-layer receiving the frame for transmission and delivering it to the physical layer. IEEE 802.11a by default does not use the RTS/CTS mechanism for reserving the wireless channel, so collision avoidance based on detecting the channel idle for a certain amount of time is used. If the wireless medium is very busy, the MAC delay will increase because the sender will not be able to detect that the channel is idle for the entire required period.

In Figures 5-10, we show the cumulative distribution functions (CDFs) of the MAC delay experienced by primary frames and aggregated frames both with and without using position verification in CASCADE. We considered three different traffic densities (low, medium and high). Using position verification with CASCADE only adds a small increase in the MAC delay. The percentage of MAC delay increase rises with increasing traffic density. This is because the position verification does not use p-IVG dissemination when sending the quarantine proposals and requests. This implies that as the traffic density increases, more messages will be sent out and the competition on the wireless channel will increase, leading to an increase in the MAC delay. Still, the increase in MAC delay is on the order of a few tens of microseconds, which should not greatly affect the operation of CASCADE.

2) Reception Rate: To assess the impact of wireless collisions, we measured the average reception rate over time. If a frame was transmitted and no other vehicle received it, then the frame was considered to not be received. Either the frame experienced a collision or no other vehicle was within 300 m of the sender.

Figures 11-13 show the reception rate of CASCADE with and without using position verification at different traffic densities (low, medium and high), averaged every 10 seconds. Using CASCADE with position verification results in slightly lower reception rate than without position verification. Again, as with MAC delay, this lower performance varies with the traffic density. The explanation for the lowered reception rate are the same as for the slightly increased MAC delay. The p-IVG dissemination technique is not used, and position verification adds

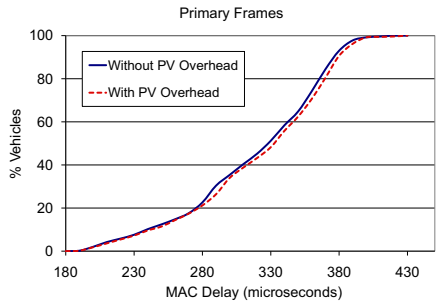

Fig. 5: MAC Delay at Low Density

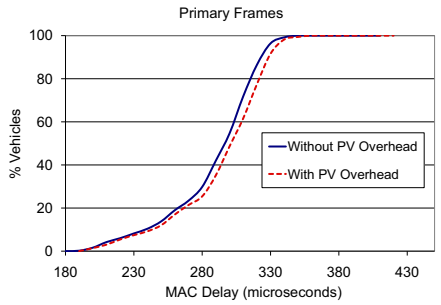

Fig. 6: MAC Delay at Medium Density

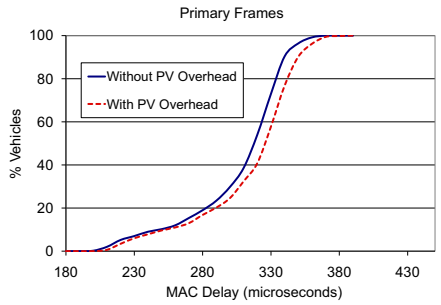

Fig. 7: MAC Delay at High Density

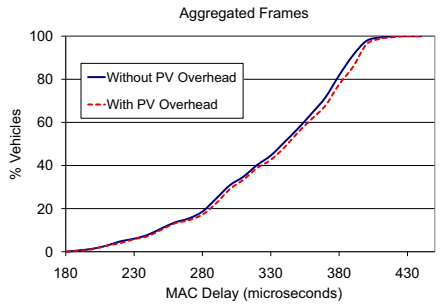

Fig. 8: MAC Delay at Low Density

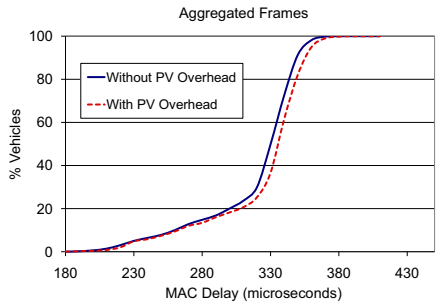

Fig. 9: MAC Delay at Medium Density

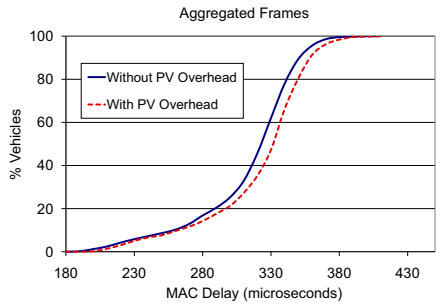

Fig. 10: MAC Delay at High Density

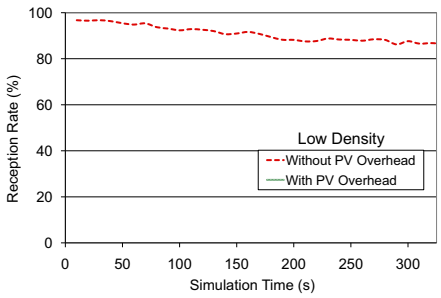

Fig. 11: Reception Rate at Low Density

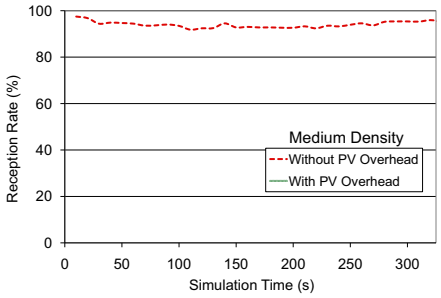

Fig. 12: Reception Rate at Medium Density

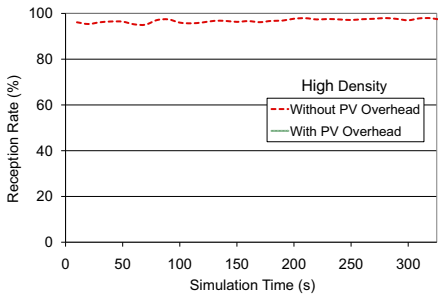

Fig. 13: Reception Rate at High Density

some additional messages that take compete on the wireless channel with the primary and aggregated frames.

3) View Update Delay: As we have seen, using the position verification technique with CASCADE adds slightly more MAC delay and reduces the reception rate. Although, the increase percentage in the MAC delay and the reception rate reduction percentage are very small, they may have an impact on the main CASCADE functionality of providing the driver with an up-to-date view of the traffic ahead. The rate at which the view is updated is a very important factor in the accuracy of the view. We measure the update delay as the time difference between receiving two consecutive aggregated frames, which is the main update source for the driver view.

Figures 14-16 show the view update delay of CASCADE with and without position verification at different traffic densities (low, medium and high), averaged every 10 seconds. These figures show that although there is a slight increase in the view update delay, the difference even with high density traffic is only about 40 ms. This amount of time is so small that the driver will not be able to notice, much less react to, the difference in update delay.

We have shown that adding position verification to CASCADE can produce great benefit in terms of security from attack with little impact on the operation of CASCADE's main purpose, alerting the driver to upcoming traffic conditions.

VII. Conclusion and Future Work

We have presented a light weight position verification technique for VANETs. Our technique used received signal strength (RSSI) and laser rangefinders for position verification. We have used the position verification technique to tighten CASCADE security and at the same time to evaluate the technique performance and its incurred overhead when applied to a particular VANET application (CASCADE). Our defense techniques can quickly detect false inputs given to the system and quarantine those vehicles responsible with very low rates of false positives. At the same time the extra overhead incurred due to applying this technique to CASCADE was negligible.

In future work, we plan to incorporate the data in aggregated frames into an additional consistency check in order to reduce the number of false positives. We also plan to take advantage of vehicles traveling in the opposite direction for position

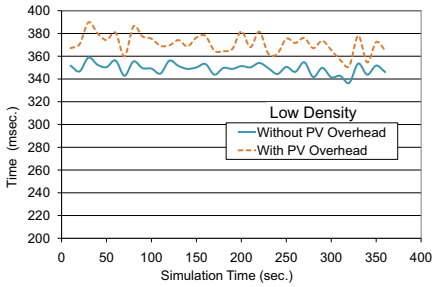

Fig. 14: The View Update Rate at Low Density

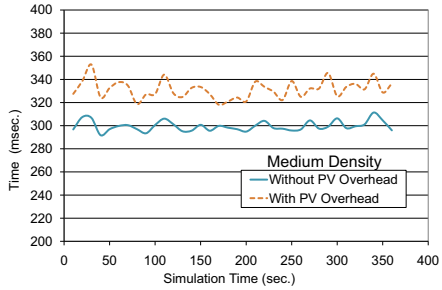

Fig. 15: The View Update Rate at Medium Density

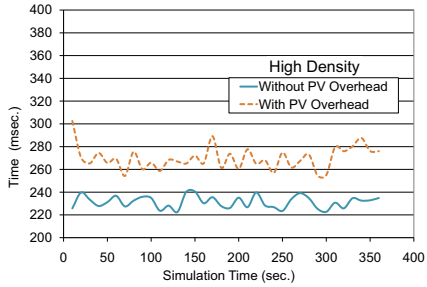

Fig. 16: The View Update Rate at High Density

verification. In addition to these benefits, we have shown that adding position verification to CASCADE has little impact on the operation of CASCADE's main purpose, alerting the driver to upcoming traffic conditions.

VIII. Acknowledgments

This work was supported by the National Science Foundation under Grant CNS-0721586.

References

[1] K. Ibrahim and M. C. Weigle, "Accurate data aggregation for VANETs," in *Proceedings of ACM VANET*, Montreal, Canada, Sep. 2007, pp. 71–72.

[2] ——, "CASCADE: Cluster-based accurate syntactic compression of aggregated data in VANETs," in *Proceedings of IEEE AutoNet*, New Orleans, LA, Dec. 2008, accepted.

[3] M. Raya and J.-P. Hubaux, "The security of vehicular ad hoc networks," in *Proceedings of the ACM SASN*, Alexandria, VA, Nov. 2005, pp. 11–21.

[4] F. Picconi, N. Ravi, M. Gruteser, and L. Iftode, "Probabilistic validation of aggregated data in vehicular ad-hoc networks," in *Proceedings of ACM VANET*, Los Angeles, CA, Sep. 2006, pp. 76–85.

[5] J.-P. Hubaux, S. Capkun, and J. Luo, "The security and privacy of smart vehicles," *IEEE Security and Privacy Magazine*, vol. 2, no. 3, pp. 49–55, 2004.

[6] M. Raya, A. Aziz, and J.-P. Hubaux, "Efficient secure aggregation in VANETs," in *Proceedings of ACM VANET*, Los Angeles, CA, Sep. 2006, pp. 67–75.

[7] M. Raya, P. Papadimitratos, and J.-P. Hubaux, "Securing vehicular communications," *IEEE Wireless Communications Magazine*, vol. 13, no. 5, pp. 8–15, Oct. 2006.

[8] S. Capkun and J.-P. Hubaux, "Secure positioning of wireless devices with application to sensor networks," in *Proceedings of the IEEE INFOCOM*, vol. 3, Miami, FL, March 2005, pp. 1917–1928.

[9] B. Parno and A. Perrig, "Challenges in securing vehicular networks," in *Proceedings of the ACM HotNets*, College Park, MD, 2005.

[10] F. Armknecht, A. Festag, D. Westhoff, and K. Zeng, "Cross-layer privacy enhancement and non-repudiation in vehicular communication," in *Proceedings of WMAN*, Bern, Switzerland, 2007.

[11] J. Y. Choi, P. Golle, and M. Jakobsson, "Tamper-evident digital signature protecting certification authorities against malware," in *Proceedings of the IEEE DASC*, Indianapolis, IN, 2006, pp. 37–44.

[12] N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims," in *Proceedings of ACM WiSe*, San Diego, CA, 2003, pp. 1–10.

[13] T. Suen and A. Yasinsac, "Peer identification in wireless and sensor networks using signal properties," in *Proceedings of the IEEE WSNS*, Washington, DC, Nov. 2005, pp. 1–8.

[14] T. Leinmüller, E. Schoch, F. Kargl, and C. Maihöfer, "Influence of falsified position data on geographic ad-hoc routing," in *Proceedings of ESAS*, 2005.

[15] T. Leinmüller and E. Schoch, "Greedy routing in highway scenarios: The impact of position faking nodes," in *Proceedings of WIT*, 2006.

[16] B. Xiao, B. Yu, and C. Gao, "Detection and localization of sybil nodes in VANETs," in *Proceedings of ACM DIWANS*, Los Angeles, CA, 2006, pp. 1–8.

[17] P. Golle, D. Greene, and J. Staddon, "Detecting and correcting malicious data in VANETs," in *Proceedings of ACM VANET*, Philadelphia, PA, Oct. 2004, pp. 29–37.

[18] G. Yan, S. Olariu, and M. C. Weigle, "Providing VANET security through active position detection," *Computer Communications: Special Issue on Mobility Protocols for ITS/VANET*, vol. 31, no. 12, pp. 2883–2897, jul 2008.

[19] K. Ibrahim and M. C. Weigle, "p-IVG: Probabilistic inter-vehicle geocast for dense vehicular networks," Old Dominion University, Tech. Rep., Sep. 2008.

[20] A. Bachir and A. Benslimane, "A multicast protocol in ad hoc networks: Inter-vehicle geocast," in *Proceedings of IEEE VTC-Spring*, Jeju Island, Korea, Apr. 2003, pp. 2456–2460.

[21] K. Ibrahim and M. C. Weigle, "Optimizing CASCADE data aggregation for VANETs," in *Proceedings of the IEEE MoVeNet*, Atlanta, GA, Sep. 2008, pp. 724–729.

[22] P. Enge, "Retooling the global positioning system," *Scientific American*, May 2004.

[23] US Department of Transportation, "Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems," ASTM E2213-03, Aug. 2003.

[24] "SICK - Sensor Intelligence," http://www.sick.com.

[25] "ibeo Automobile Sensor," http://www.ibeo-as.com.

[26] R. A. MacLachlan and C. Mertz, "Tracking of moving objects from a moving vehicle using a scanning laser rangefinder," in *Proceedings of IEEE ITSC*, Toronto, Canada, Sep. 2006, pp. 301–306.

[27] F. Fayad and V. Cherfaoui, "Tracking objects using a laser scanner in driving situation based on modeling target shape," in *Proceedings of the IEEE IV Symposium*, Istanbul, Turkey, Jun. 2007, pp. 44–49.

[28] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, Eds., *Mobile Ad Hoc Networking*. Wiley-IEEE Press, Dec. 2003.

[29] S. Saha, K. Chaudhuri, D. Sanghi, and P. Bhagwat, "Location determination of a mobile device using IEEE 802.11b access point signals," in *Proceedings of IEEE WCNC*, New Orleans, LA, Mar. 2003, pp. 1987–1992.

[30] R. Miller and Q. Huang, "An adaptive peer-to-peer collision warning system," in *Proceedings of IEEE VTC-Spring*, Birmingham, AL, May 2002, pp. 317–321.

[31] K. Ibrahim and M. C. Weigle, "ASH: Application-aware SWANS with highway mobility," in *Proceedings of IEEE MOVE*, Phoenix, AZ, Apr. 2008.

[32] "JiST/SWANS," http://jist.ece.cornell.edu, 2004.

[33] R. Barr, Z. Hass, and R. van Renesse, *Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad hoc Wireless, and Peer-to-Peer Networks*. CRC Press, 2005, ch. 19: Scalable Wireless Ad Hoc Network Simulation, pp. 297–311.

[34] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Review E*, vol. 62, no. 2, pp. 1805–1824, 2000.

[35] A. Kesting, M. Treiber, and D. Helbing, "MOBIL: General lane-changing model for car-following models," in *Proceedings of the Transportation Research Board Annual Meeting*, Washington, DC, Jan. 2007.

[36] J. Yin, T. ElBatt, G. Yeung, B. Ryu, S. Habermas, H. Krishnan, and T. Talty, "Performance evaluation of safety applications over DSRC vehicular ad hoc networks," in *Proceedings of ACM VANET*, Philadelphia, PA, Oct. 2004, pp. 1–9.

[37] N. Wisitpongphan, F. Bai, P. Mudalige, V. Sadekar, and O. Tonguz, "Routing in sparse vehicular ad hoc wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1538–1556, Oct. 2007.

[38] "Berkeley highway lab (BHL)," http://bhl.calccit.org/, 2006.