CASCADE: Cluster-based Accurate Syntactic Compression of Aggregated Data in VANETs

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

***Abstract*—We present a method for accurate aggregation of highway traffic information in vehicular ad hoc networks (VANETs). Highway congestion notification applications need to disseminate information about traffic conditions to distant vehicles. In dense traffic, aggregation is needed to allow a single frame to carry information about a large number of vehicles. Our technique, CASCADE, uses compression to provide aggregation without losing accuracy. We show that CASCADE makes efficient use of the wireless channel while providing each vehicle with data that is highly accurate, represents a large area in front of the vehicle, and can be combined with aggregated data from other vehicles to further extend the covered area.**

I. Introduction

Highway traffic congestion is costly. Estimates have shown that millions of hours and billions of dollars are wasted each year because of congested roadways [1]. Advance notification of traffic congestion would allow many drivers to take alternate routes and save time. Vehicular Ad-hoc Networks (VANETs) have been proposed as a means to provide this advance notification of traffic congestion to drivers. A 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 provide information to non-neighboring vehicles, 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. In order to reduce the number of messages that need to be sent, several data aggregation techniques have been proposed [2]–[7]. Unfortunately, with these techniques, some accuracy of the data is lost upon aggregation, and data aggregated by one vehicle cannot be combined with data aggregated by another.

In this paper, we present CASCADE (*Cluster-based Accurate Syntactic Compression of Aggregated Data in VANETs*), a new method for accurate aggregation of traffic information in VANETs, featuring cluster-based compression. In aggregated frames, we represent each vehicle's location based on its difference from the location of the center of the cluster and its speed based on its difference from the median speed of all vehicles in the cluster. In this way, accurate information can be distributed in a small number of bytes. CASCADE uses *probabilistic Inter-Vehicle Geocast* (p-IVG) [8], a modification to Inter-Vehicle Geocast [9] that adapts the re-broadcasting of frames based on the surrounding traffic density for more efficient use of the wireless channel. CASCADE is designed to enable both safety (collision warning) and information (congestion notification) applications.

We show that CASCADE makes efficient use of the wireless channel while providing vehicles with data that is highly accurate, represents a large area in front of the vehicle, and can be combined with aggregated data from other vehicles to further extend the covered area.

The remainder of the paper is organized as follows. In Section II we present an overview of related work in data aggregation in wireless and vehicular networks. In Section III we describe the CASCADE system in detail. We analyze the system in Section IV and present results of our simulation studies in Section V. Finally, we conclude with a summary and our plans for future work in Section VI.

II. Related Work

Data aggregation has received much attention in the wireless sensor network community [10]–[12], but many of the approaches either assume a static network or require several rounds of communication between nodes to provide security. Both of these requirements are impractical for VANETs.

There has been recent work on data aggregation techniques specifically designed for VANETs. Picconi *et al.* [3] classified aggregation techniques as either *syntactic* or *semantic*. Syntactic aggregation uses a technique to compress or encode the data from multiple vehicles in order to fit the data into a single frame. This results in lower overhead than sending each message individually. In semantic aggregation, the data from individual vehicles is summarized. For instance, instead of reporting the exact position of five vehicles, only the fact that five vehicles exist is reported. The trade-off is a much smaller message in exchange for a loss of precise data.

Nadeem *et al.* [2] present the *TrafficView* system, which uses semantic aggregation. The authors present two techniques for aggregation: *ratio-based* and *cost-based*. In the ratio-based technique, the roadway in front of a vehicle is divided into regions. Data is aggregated based on ratios that have been pre-assigned to each region. Regions farther away from a vehicle are assigned larger aggregation ratios, because precise detail may not be needed over a long range. The resulting view of traffic conditions is, thus, customized for each particular vehicle. For this reason, the produced view may not be useful for other vehicles unless they use the same aggregation ratios. In the cost-based aggregation technique, data is aggregated based on a cost function that depends on the position of the aggregating vehicle. For this reason, the produced view of the traffic is not useful to any other vehicle unless it is close to the aggregating vehicle.

Lochert *et al.* [13] present a probabilistic technique for aggregating the disseminated data in VANET applications. The proposed technique does not aggregate the actual values but uses a modified Flajolet-Martin sketch as a probabilistic approximation for the values. This technique can be applied to aggregate the data in any non-accuracy-sensitive application (*e.g.*, estimating the number of available parking spaces), but it cannot be used in our target application, which requires the actual vehicle information to be disseminated and re-aggregated to reach distant vehicles.

Yu *et al.* [6] present an aggregation technique called Catch-Up that aggregates similar reports generated by the vehicles whenever an event occurs *e.g.*, a change in vehicle's density. The technique is based on inserting a delay before forwarding any report in the hopes of receiving similar reports from surrounding vehicles so that these reports can be aggregated into a single report. Since Catch-Up inserts a delay before forwarding messages, it would not be suitable for safety applications, such as collision warning.

Lochert *et al.* [5] describe a hierarchal aggregation technique for vehicle travel times. In this technique each vehicle broadcasts its travel time between two landmarks along its trip. Then these travel times are aggregated hierarchically and broadcasted to provide distant vehicles with an estimate of the travel times along the road segments so that they can avoid congested roads (the roads with larger travel time estimate). In case of a slow driver traveling along the road, the vehicle will report a long travel time between any two landmarks, which will be translated by the other vehicles as congestion between those landmarks even though there may be no congestion on the roadway. As with Catch-Up, this work does not disseminate or aggregate information suitable for safety applications, but is only concerned with reporting traffic conditions.

Saleet *et al.* [7] present a location query protocol that aggregates data in VANETs. The protocol divides the road in to segments and the closest node to the segment center plays the server role. Each vehicle periodically broadcasts its information, and the server node is responsible for storing this information, aggregating it, and then broadcasting it. The aggregation technique is different than ours in that it does not include any compression mechanisms. Also, the data dissemination is based on pure flooding and targets the local area only while ours is based on GeoCast and adapts itself based on the traffic density. Moreover, the main target is to provide a location query facility which is orthogonal to our target applications, which are collision avoidance and congestion notification.

III. CASCADE

The goal of CASCADE is to allow a vehicle to obtain an accurate view of upcoming traffic conditions. Vehicles will pass information about traffic conditions ahead of them to vehicles behind them so that these vehicles will have timely notification of upcoming traffic conditions. CASCADE can also support cooperative collision warning applications [14], and so, vehicles exchange their position information with neighboring vehicles several times a second.

A. Assumptions

We assume that each vehicle in the system is equipped with a GPS receiver for obtaining location and time, a navigation system that can map GPS coordinates to a particular roadway and offer routes. The GPS precision is in the order of meters, which will result in inaccurate position estimation. Recently most of the new GPS receivers support the differential correction technology (DGPS), which reduces the error in estimating positions to the order of centimeters [15]. Each vehicle is also equipped with a communications device using Dedicated Short Range Communications (DSRC) [16]. DSRC, with a transmission rate of 6-11 Mbps, is based on the upcoming IEEE 802.11p standard, which is a part of the larger IEEE 1609 Family of Standards for Wireless Access in Vehicular Environments (WAVE) [17]. Each vehicle is also pre-assigned a public/private key pair and the public key's certificate, used for authentication. To address privacy concerns, each vehicle may also use multiple pseudonyms to disguise its public keys [18], [19]. In our design of CASCADE, we assume a four-lane highway with 4 m wide lanes and ignore vehicles traveling in the opposite direction, although these vehicles may be used to disseminate reports during sparse traffic conditions.

B. Overview

Before describing the details of the system, we present a brief high-level overview. Each vehicle periodically broadcasts its position information (including location, speed, acceleration, and heading), which we call a *primary record*. Received primary records are stored in a local database in each vehicle. Those primary records representing vehicles ahead of the current vehicle comprise the *local view*. The local view, as shown in Figure 1, is divided into *clusters*[1]. Each cluster has a width of 16 m (4 lanes) and a length of 126 m, set for maximal record compression (to be described in Section III-D1). Selecting the cluster dimensions 16 m x 126 m is based on optimal cluster size analysis, balancing the trade-off between local view length and expected frame size [20]. There are 12 rows of clusters in a local view, resulting in a visibility of 1.5 km (exactly 1512 m).

As the local view is longer than the typical DSRC transmission range (about 300 m [21]), primary records may be

[1]The term *cluster* is used here in local sense only. Each vehicle will assign vehicles it knows about into the appropriate cluster based on the vehicles' distances from itself. Thus, there is no need for cluster management or node agreement on which vehicles are in which clusters.

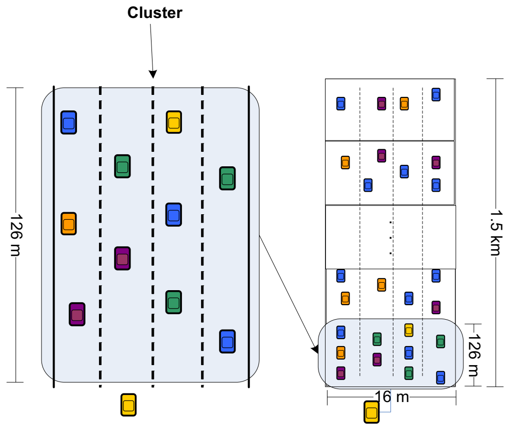

Fig. 1: Vehicle's Local View, Divided into 16 m x 126 m Clusters

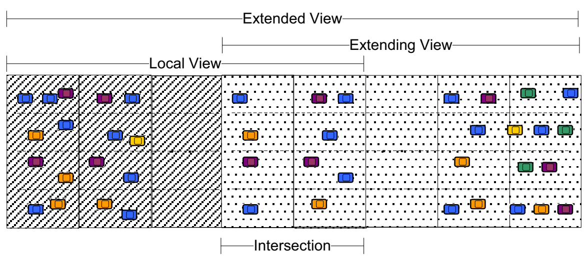

Fig. 2: Vehicle's Local View and Extended View

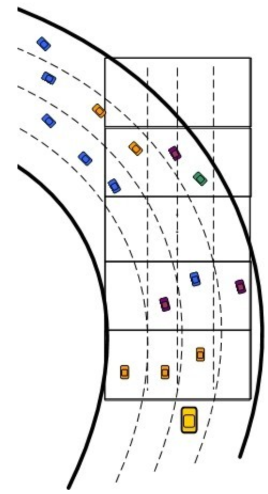

Fig. 3: Vehicle's Local View Mapped onto a Curved Roadway

re-broadcast (to a maximum of 1.5 km behind the original sender).

Each vehicle periodically compresses and aggregates the primary records in its local view into an *aggregated record*. This aggregated record is then broadcast to neighboring vehicles. Received aggregated records may be used to augment the local view by providing information about vehicles beyond the local view, resulting in an *extended view*. Figure 2 shows an example of the local view and extended view.

Although our examples feature a straight, rectangular-shaped road, CASCADE is not limited to such geometries. Figure 3 shows a local view mapped onto a curved roadway. As the vehicle enters the curve, more of the vehicles inside and past the curve will be added to its local view.

C. Primary Records

A vehicle's local view is built entirely of received primary records. The primary record contains the basic information for a single vehicle. Each record can be represented in 29 bytes:

- *timestamp* (8 bytes) - time the record was generated
- *location* (16 bytes) - latitude and longitude
- *speed* (1 byte) - in meters/second
- *acceleration* (1 byte) - in meters/second^2
- *heading* (1 byte) - in degrees from North (0-360)
- *altitude* (2 bytes) - in meters above sea level

1) Initial Dissemination: A vehicle broadcasts a *primary frame* containing its primary record at a random interval between 300-400 ms, which is consistent with message frequency recommendations for collision warning applications [22]. The primary frame, totaling 1033 bits[2], consists of:

- *type* (1 bit) - primary or aggregated frame
- *sender's location* (16 bytes) - latitude/longitude
- *primary record* (29 bytes)
- *digital signature* (28 bytes)
- *certificate* (56 bytes)

The sender's location in the primary frame is updated each time the primary frame is re-broadcast by another vehicle. The primary record is signed by the original vehicle using ECDSA [23]. The certificate included in the frame contains the original vehicle's public key, signed by the certificate authority. Since the primary record is signed by the original sender, it cannot be tampered with by a re-broadcasting node without detection. In addition, replay attacks are nullified by the presence of the timestamp inside the signed primary record.

A receiving vehicle will record the primary record and use the vehicle's public key as an identifier. Typically, only primary records from vehicles within the receiving vehicle's local view (*i.e.*, vehicles in front of the receiving vehicle) will be stored. But for some applications, such as merging assistance, awareness of vehicles behind or beside the receiving vehicle is important. In these cases, the vehicle would store the primary records of nearby (within one cluster, or 126 m) following vehicles to be used in the application. Again, since these records are from following vehicles, they are not considered part of the local view.

2) Re-Broadcast: In order for primary records to reach vehicles farther than 300 m, the records must be re-broadcast. In order to limit the number of re-broadcast messages used to propagate the frames, we use an adaptation of the Inter-Vehicle Geocast (IVG) algorithm [9] to accommodate the high volume of frames that need to be re-broadcast. In IVG, each node starts

[2]If the underlying link-layer requires a frame size of full bytes, then the primary frame would be padded to 130 bytes.

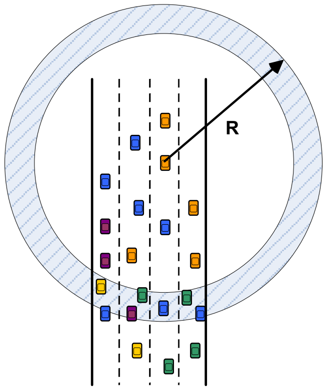

Fig. 4: Nearby Vehicles at the Boundary

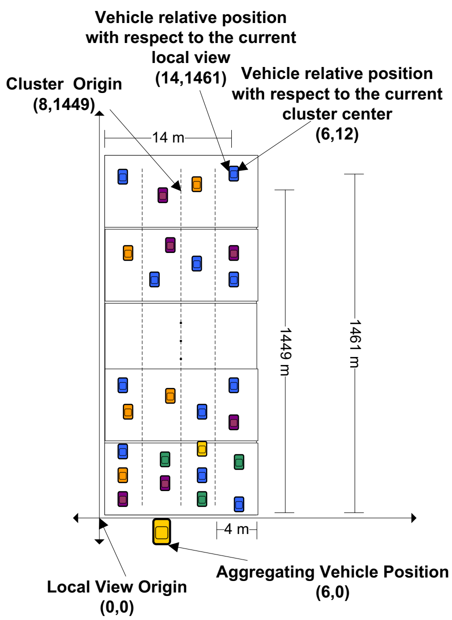

Fig. 5: Converting GPS Coordinates to {X,Y} Coordinates

a timer for each frame it receives. If the timer expires and the frame associated with this timer has not been re-broadcast by any other node, the node re-broadcasts the frame. The timer value T_x for vehicle x is

$$T_x = T_{max} \cdot \frac{(R^\epsilon - D_{sx}^\epsilon)}{R^\epsilon},$$

where R is the transmission range and D_{sx} is the distance between vehicle x and vehicle s, the sender of the message, and $\epsilon = 2$ to generate a uniform timer value between [0, T_{max}], where $T_{max} = 200$ ms [9].

By using IVG with its original settings, nodes that are close to each other at the boundary will have very similar, if not equal, timer values. This means the nodes' timers will expire at almost the same time, and the nodes will rebroadcast the frame essentially simultaneously, resulting collisions. This situation is shown in Figure 4, where the highlighted nodes, being a similar distance from the sender, will have similar timer values.

To alleviate this problem, we modified IVG by making the re-broadcasting of frames probabilistic based on the surrounding vehicle density. In *probabilistic-IVG* (p-IVG) [8], when a vehicle receives a frame, it first selects a random number in [0,1]. If the selected number is less than $\frac{1}{density}$, the timer is started. If not, then the frame will not be re-broadcast by this vehicle. As the density increases, the number of nodes that will start their timers decreases. Since each vehicle periodically broadcasts its primary frame, at any time a vehicle knows the location of other vehicles within its transmission range. The density is then calculated as the number of vehicles present divided by the size of the area of interest. Because primary frames will be broadcast by the original sender every 300 ms, there is ample opportunity for fresh frames to be re-broadcast even if all vehicles discard a message during a particular round. In addition to adding the probabilistic timer, we use $\epsilon = 0.5$ in order to produce sparser timer values, because the timer values decrease faster as the distance from the original sender increases.

In addition to using p-IVG to limit the number of re-broadcasts, primary frames have a time-to-live (TTL) value to ensure that only fresh information is disseminated. The TTL of all primary frames in CASCADE is 1 second. If a node receives a frame and the difference between the original sending time and the current time is greater than the primary frame TTL, the frame will be dropped.

3) Aging: The goal of CASCADE is to present highly accurate information about upcoming traffic conditions. So, it is important that old information is purged from the system. As the local view is concerned only with vehicles in front of the current vehicle, primary records are removed from the local view (but not necessarily from a vehicle's database) once the vehicle has physically passed the vehicle described by the record. Additionally, primary records may also be removed from the local view when no updates have been received in 1 second. With vehicles sending new primary frames 3 times a second, receiving no message about a vehicle for 1 second means that 3 messages in a row were not received, which would indicate that the vehicle corresponding to the old record has likely left the area.

D. Aggregated Records

Each vehicle builds its local view based on primary records received from other vehicles. In order to extend the view farther, vehicles exchange aggregated records. Here, we describe how primary records are grouped into clusters, how clusters are aggregated, and how the aggregated records are disseminated and used to build the extended view.

1) Compression: As primary records are received, the vehicles described in those records are grouped into their corresponding clusters, based on their distance from the receiving vehicle. When clustering is done, a vehicle's heading and altitude are taken into account to ensure that vehicles are assigned to the proper cluster.

The *compact data record* is used to represent a single vehicle within a cluster. The compression is achieved by using a variation on differential coding that is more efficient in compressing the vehicular data. CASCADE represents only the differences between the vehicle data and overall cluster data. Before the compact data record is formed, the median speed of all vehicles in a cluster is calculated, the position of the center of the cluster is calculated, and the position of each vehicle is translated into $\{X, Y\}$ coordinates (in integer meters) with the local view origin as the origin, as shown in Figure 5. For this, we assume that the digital map in the vehicle provides the GPS position of the leftmost lane of the roadway.

Each compact data record, totaling 19 bits, contains the following fields:

- ΔX (5 bits) - difference between the vehicle's X coordinate and the X coordinate for the center of its cluster
- ΔY (7 bits) - the difference between the vehicle's Y coordinate and the Y coordinate for the center of its cluster
- ΔS (5 bits) - the difference between vehicle's speed and the median speed of the vehicles in the cluster
- *Speed Indicator (SI) Flag* (2 bits) - indicates if the vehicle's speed is within the acceptable range for the cluster

Since the base value is the center of the cluster (as in Figure 5), ΔX can have a negative value. With sign-magnitude representation, this means that five bits must be used for ΔX, one for the sign and four for the magnitude (with a maximum width difference of 8 m).

Since the cluster length is 126 m, ΔY can be represented with only 7 bits, using 1 bit for the sign and 6 bits for the difference between the vehicle's position and the cluster center (at most 63 m).

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*

The *SI Flag* has an important application in collision warning. Many accidents are due to vehicles that are either traveling much faster than surrounding vehicles (*speeders*) or traveling much slower than surrounding vehicles (*laggers*). If drivers can be alerted to these vehicles in advance, they may be able to avoid accidents.

With CASCADE, we achieve a compression ratio of at least 86%. The primary data for each vehicle (location and speed) is represented in 136 bits (17 bytes) while the compact data for each vehicle is represented in at most 19 bits. The compression ratio is even higher if the ΔS field is omitted, as in the case of speeders and laggers.

E. Aggregation

Once compression has been completed, we form an *aggregated cluster record*, which is a concatenation of the compact data records of the vehicles in the cluster. Each aggregated cluster record 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* (7 bits) - the number of vehicles contained in the cluster
- *compact data records* (19 bits each) - concatenation of all of the compact data records for vehicles in this cluster

If there is a cluster that contains no vehicles, its *cluster flag* is set to 0 and no more information about the cluster is contained in the record.

Since a cluster is four lanes wide and 126 m long, and the average vehicle length is 5 m with an inter-vehicle distance of 2 m, there can be at most 72 vehicles in a cluster, which can be represented in 7 bits.

1) Initial Dissemination: Every 4 seconds, compression and aggregation is done. Once the aggregated cluster records are constructed, they are concatenated into a single frame and sent via broadcast. The aggregated frame includes the following fields:

- *type* (1 bit) - primary or aggregated frame
- *timestamp* (8 bytes)
- *aggregating vehicle's X-coordinate* (5 bits) - meters from the vehicle's local view origin, assuming 4 lanes of traffic
- *aggregating vehicle's location* (19 bytes)
- *aggregated cluster records* - up to 12 records
- *digital signature* (28 bytes)
- *certificate* (56 bytes)
- *sender's location* (16 bytes) - latitude/longitude

The signature is calculated by the aggregating vehicle over all the fields in the aggregated frame except the certificate which is signed by the certificate authority (CA) and the sender's location, which represents the location of the last vehicle that broadcast the frame.

Note that if traffic is sparse, the aggregated frame will be much smaller than the maximum 2312 bytes, because empty clusters are represented by a single bit.

The cluster aggregated records are arranged according to their place in the view, starting with the bottom-left cluster, moving from left to right, and then increasing in distance from the aggregating vehicle. Since these records are always arranged in the same manner, there is no need for a cluster ID to be included in the frame or record.

2) Re-Broadcast: Aggregated frames are re-broadcast in the same manner as primary frames, using the p-IVG algorithm described in Section III-C2. Aggregated frames originating from vehicles physically behind the receiving vehicle will be dropped, as well aggregated frames that are older than the TTL. All aggregated frames have a TTL of 2 seconds, balancing timeliness of the data and the distance that the aggregated frame could travel (see Section IV). As with primary frames, the re-broadcaster will update the sender's location field in the aggregated frame before transmitting.

F. Building the Extended View

When a vehicle receives an aggregated frame, it first checks to see if the aggregating vehicle's position is within its current view. If so, then there must be an overlap between the receiving vehicle's view and the view contained in the aggregated frame (Figure 2). The receiving vehicle reconstructs the primary data for the vehicles in the received aggregated frame. Before placing the vehicles in the view, their positions are adjusted based on the speed they were traveling and the time since the aggregated frame was broadcast. Once adjusted, the receiving vehicle compares the new vehicle data in the intersecting area with the data already in its current view. A number of vehicles will overlap (*i.e.*, be very close together and traveling at similar speeds). If over 75% of the vehicles in the intersecting region overlap, then the received view is declared to be consistent with the current view, so the non-intersecting part in the received view can be used to extend the current view. Once the current view has been extended, this view can be extended further through the receipt of other aggregated frames. Information about all vehicles not in the local view has a lifetime of only 10 seconds. Vehicles will be periodically receiving new aggregated frames that can be used to re-build the extended view with fresh data.

The comparison between the vehicle's data in the intersecting area can be used for detecting malicious vehicles that try to inject false views in the traffic. This problem has been addressed before [24], but requires more overhead and processing. In our approach, these vehicles can be caught at no extra cost because the comparison of the intersecting views are an essential part to extending the view. This procedure can be enhanced in future work with a framework to disseminate the identity of lying vehicles and isolate them.

IV. Analysis

Table 1 lists some basic constants that will be used in the analysis of CASCADE.

A. Visibility

The maximum visibility with the local view is 1.5 km. We derive that value here. The maximum visibility is based on the maximum number of clusters, N_C, that can fit into a single MAC layer frame, the maximum length of each cluster, L_C, and the number of lanes on the road, N_L. The number of vehicles per cluster, N_{VPC}, depends on the size of the cluster, the width of a lane, the length of a vehicle, and the distance between vehicles.

$$N_{VPC} = \frac{L_C W_C}{(L_V + D) W_L},$$

where D is the average distance between vehicles. In the case of dense traffic, where D can be as little as 2 m, the maximum number of vehicles per cluster is 72 vehicles.

The number of clusters that can be transmitted in a single frame, N_{CPF}, is determined by the size of the frame, the header and encryption data needed for the aggregated frame, and the size of the aggregated cluster record. Several of these quantities are listed in Table I.

$$N_{CPF} = \frac{S_F - H_{AF} - E_{AF}}{S_{ACR}}$$

The size of an aggregated cluster record, S_{ACR}, is determined by the number of vehicles in the cluster, the size of the aggregated vehicle record, and the amount of header information needed for the aggregated cluster record.

$$S_{ACR} = (S_{CDR} N_{VPC}) + H_{ACR}$$

So, in dense traffic, which would be the worst case, N_{VPC} is 72 and S_{ACR} is 1384 bits. So, at most 12 clusters can fit into a single frame.

If we assume that on average, there are 4 lanes on the highway, these 12 clusters would be divided into 12 cluster rows. Since each cluster represents 126 m ahead, the visibility of these 12 cluster rows is 1512 m, or 1.5 km.

For the extended view, we add local views from other vehicles to the base vehicles' local view. To determine the maximum visibility for an extended view, recall that aggregated frames, containing a vehicle's entire local view, have a lifetime limited to 2 seconds after they were originally sent. In dense traffic, frames will be the maximum 2312 bytes long. For DSRC with a 6 Mbps data rate, the transmission time of such a frame is about 3 ms. Also, in dense traffic, each transmission will reach a vehicle 300 m away before being re-broadcast. The propagation delay for 300 m is about 1 μs, which we treat as negligible. Assuming a conservative 20 ms processing delay at each hop, including medium access delays, a full frame can travel 300 m in about 23 ms. So, in 2 seconds, an aggregated frame could travel 26 km, or about 16 miles.

B. Compression

In CASCADE, we use a variation of differential coding to compress the vehicles' data in the local view in order to form a small aggregated frame. To evaluate how well CASCADE compression works, we compare it with lossless Deflate compression [25] and differential coding without using clusters. Deflate combines the LZ77 algorithm, which replaces a repeated pattern with a pointer to the first occurrence of the pattern, with Huffman coding, which assigns a shorter code for more frequently-used symbols. Deflate will work best on data that has many values in common. For differential coding, we took the first vehicle in the local view and calculated the difference between its' properties and the remaining vehicles in the local view. The main difference is that the data fields in the aggregated record will be longer because the differences will be the maximum of the local view size rather than the maximum of the cluster size.

The aggregated frame size resulting from CASCADE depends upon both the number of vehicles in the local view and on the distribution of the vehicles within the local view clusters [20]. For CASCADE, we consider the worst case vehicle distribution over the local view clusters, resulting in the maximum aggregated frame size (CASCADE-Max).

symbol	description	value
L_C	cluster length	126 m
W_C	cluster width	16 m
L_V	average vehicle length	5 m
W_L	lane width	4 m
S_{CDR}	size of one compact data record	19 bits
H_{ACR}	header for an aggregated cluster record	16 bits
H_{AF}	header for an aggregated frame	326 bits
E_{AF}	encryption trailer for an aggregated frame	672 bits (84 bytes)
S_F	IEEE 802.11p frame size	18,496 bits (2312 bytes)

TABLE I: Constants Used in Analysis

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

TABLE II: Simulation Settings

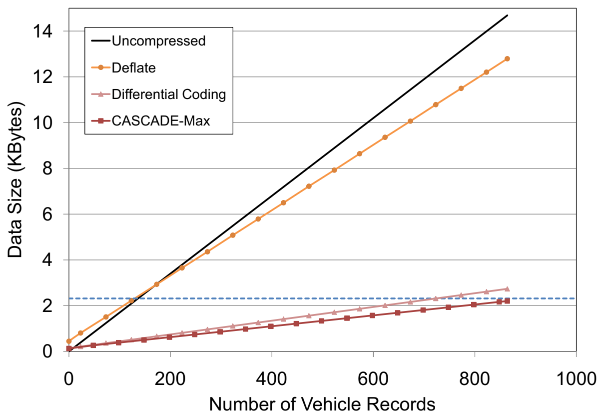

Fig. 6: Aggregated Frame Sizes Generated Using Different Compression Techniques

Figure 6 shows the aggregated frame sizes for uncompressed, Deflate, differential coding, and CASCADE-Max as the number of vehicles in the local view increases. The dotted line represents the maximum MAC-layer frame size, which is 2312 bytes. The X-axis value at the intersection point between this dotted line and each of the compression techniques curves represents the maximum number of vehicles that can be represented in a single frame. CASCADE-Max provides the best performance, allowing for 864 vehicles to be represented in a single frame, while differential coding can represent 723 vehicles. Deflate has similar performance as the uncompressed case, only representing 129 vehicles in a single frame.

Each compression technique has overhead. For small data sizes, the sum of the compressed data size and the overhead is often larger than the uncompressed data size. Further, the performance of a compression technique depends on the characteristics of the data that is being compressed. The main characteristic of the data in the local view is that the values are similar. That is why differential coding and CASCADE produce much better compression than Deflate, which depends on repeated, rather than similar, values.

V. Evaluation

A. Simulation Setup

We evaluate the feasibility of CASCADE using ASH (Application-aware SWANS with Highway mobility) [26], which is an extension of the SWANS (Scalable Wireless Ad hoc Network Simulation) vehicular network simulator [27], [28]. SWANS has been shown to be scalable and efficient, supporting large numbers of mobile nodes [29]. SWANS fully implements the IEEE 802.11a protocol, which we use as an approximation to IEEE 802.11p, and supports general mobility models, such as Random Walk and Random Waypoint. Since ASH is based on SWANS, it inherits all of its properties. Moreover, to produce realistic simulations, ASH includes implementations of the IDM (Intelligent Driver Model) vehicular mobility model [30] and the MOBIL (Minimizing Overall Braking decelerations Induced by Lane changes) lane changing model [31].

Table II summarizes our simulation settings. All vehicles in our simulations have a transmission range of 300 m [21]. The roadway used is a four-lane divided 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 CASCADE with three different traffic density scenarios. In the high density case, there are an average of 90 vehicles/km. In medium density traffic, there are an average of 66 vehicles/km, and in low density, there are an average of 53 vehicles/km. These densities were gathered from the speed and traffic volume analysis performed by Wisitpongphan *et al.* [32] for the data collected by the Berkeley Highway Lab for traffic on eastbound I-80 on June 27, 2006 [33]. In each of the three scenarios, all the vehicles in the simulation are running an application that implements CASCADE. To show the improvement provided by CASCADE data compression, we ran experiments where CASCADE either compressed the vehicles' data or sent the vehicles' data uncompressed.

B. MAC Delay

In order to investigate the impact of using compression 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

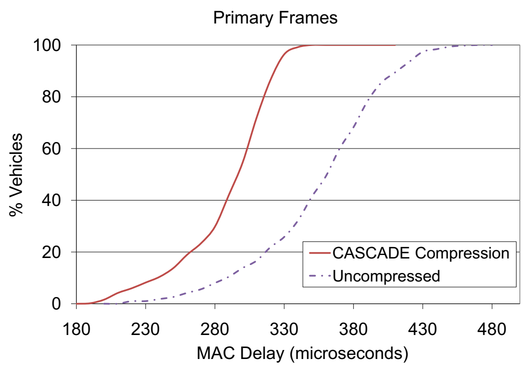

Fig. 7: MAC Delay for Primary Frames at Medium Density

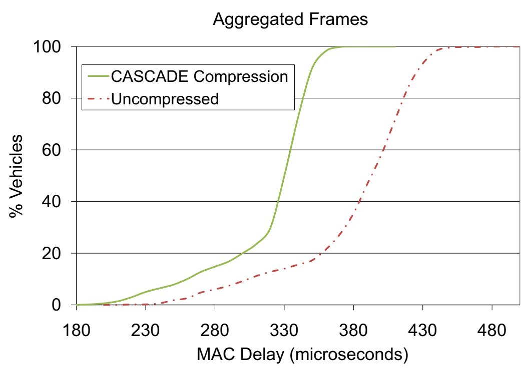

Fig. 8: MAC Delay for Aggregated Frames at Medium Density

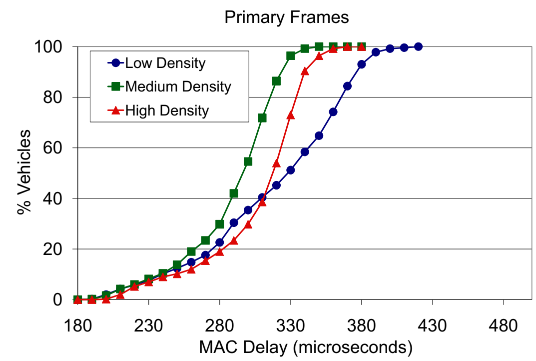

Fig. 9: MAC Delay for Primary Frames with CASCADE

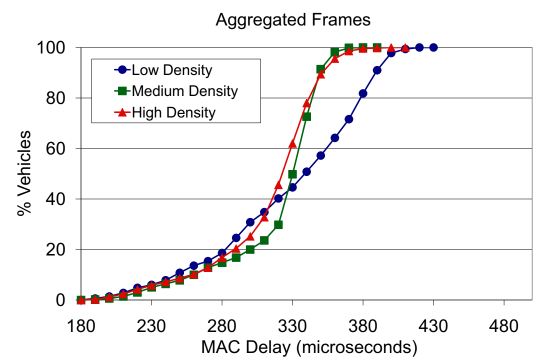

Fig. 10: MAC Delay for Aggregated Frames with CASCADE

is idle for the entire required period. We show this MAC delay as an approximation to the number of wireless collisions, which we cannot directly measure. In Figures 7 and 8, we show the cumulative distribution functions (CDFs) of the MAC delay experienced by primary frames and aggregated frames, respectively, in case of using compression or sending the vehicles' data with no compression at medium traffic density.

Both primary frames and aggregated frames see about 25% less MAC delay when using compression. This is because CASCADE compression provides smaller aggregated frames that will take less transmission and propagation time, thus making the wireless channel more available for both primary and aggregated frames.

Figures 9 and 10 show the CDFs of MAC delay experienced by primary frames and aggregated frames, respectively, when using CASCADE considering different traffic densities (low, medium and high). The maximum MAC delay is still relatively small even at high densities. The key point is that with CASCADE, which uses p-IVG, higher density does not necessarily mean more frames are sent. Density affects the re-broadcasting of frames, so at higher densities, fewer vehicles will actually re-broadcast frames.

C. 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. Figure 11 shows the reception rate of CASCADE when using either compression or no compression at medium traffic density, averaged every 10 seconds.

Using CASCADE with compression results in at least a 45% higher reception rate than using CASCADE with no data compression. That is due to the large aggregated frame size generated when no compression is used, which increases the collision probability and reduces the reception rate.

In Figure 12, we show the reception rate for CASCADE with each of the three traffic densities. In our definition, if just one vehicle receives the frame, it is considered received. High density traffic provides the best reception rate because there are more vehicles in range, thus more of a chance to receive the frame. The decreasing reception rate in the low density case is an artifact of how vehicles enter the simulation. Vehicles enter the highway with a low speed and gradually increase towards the maximum of 30 m/s. Until the last vehicle reaches the maximum speed, vehicles in front of it will be traveling faster, thus moving out of its transmission range. For low density, it takes longer for 500 vehicles to enter the system than with high density, so it takes longer for the last vehicle to reach its maximum speed.

D. Throughput

Here we investigate the bandwidth usage of CASCADE by calculating the throughput for each vehicle in the system. Throughput is defined as the total number of bits sent or

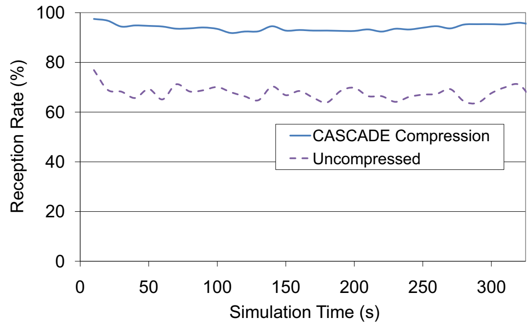

Fig. 11: Reception Rate at Medium Density

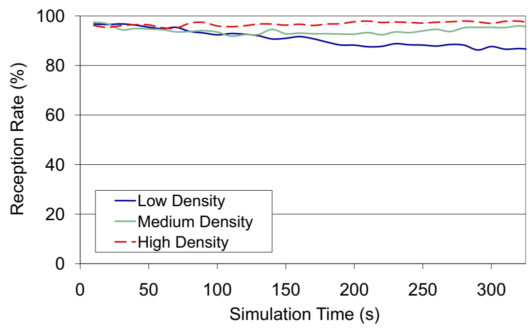

Fig. 12: Reception Rate with CASCADE

received per second. In our simulation, as each vehicle passed a certain point in the highway, its throughput was observed for the next 10 seconds and then averaged.

Table III shows the minimum, maximum, average and median throughput for the 500 vehicles in the simulation at medium traffic density. The average vehicle throughput, 76.79 kbps, can be divided into 10.40 kbps data sent and 66.39 kbps data received. CASCADE's link utilization can be approximated as the following. With medium density, an average of 18 vehicles can exist within 300 m. Within this 300 m, each vehicle is sending data at a rate of 10.40 kbps on average. This data is received by other vehicles within the range. Some of the data received by vehicles may come from vehicles outside a particular 300 m range (*i.e.*, vehicles within range of some but not all vehicles in the 300 m range we are considering). So we define effective throughput as the sum of the average rate of data sent by each vehicle and the average rate of data received divided by the number of vehicles within the transmission range. In this case, the average effective throughput for each vehicle will be 14.08 kbps. So, the total bandwidth consumed by those 18 vehicles is 253.59 kbps. As DSRC has a possible bandwidth range 6-11 Mbps, at the lowest end (6 Mbps), channel utilization is only 4.22%.

E. Visibility

The goal of CASCADE is to provide information about upcoming vehicles, so visibility is one of the most important metrics we can measure. We consider visibility to be the distance between a vehicle and farthest vehicle in its extended view. Thus, as a vehicle's visibility increases, its knowledge about upcoming traffic conditions increases.

minimum	83.61 kbps
maximum	69.22 kbps
average	76.79 kbps
median	76.58 kbps

TABLE III: Throughput

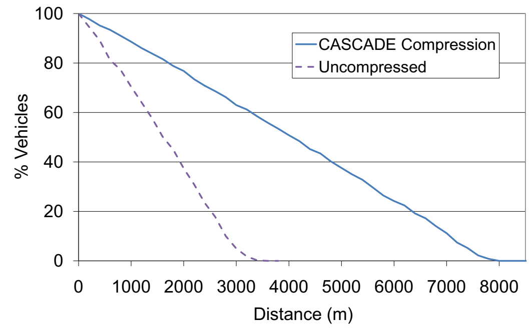

Fig. 13: Visibility at Medium Density

Figure 13 shows the percentage of vehicles that have a particular minimum visibility when using either compression or no compression at medium density traffic. Using CASCADE with compression increased the vehicles' visibility with almost 100% over when no compression is used. For example, with CASCADE compression 60% of the vehicles have a visibility of 3000 m or more, while with no compression, almost no vehicles have that much visibility. Visibility highlights the importance of the reception rate. If there are many collisions, then information is not able to be disseminated to distant vehicles, so their visibility is reduced.

Figure 14 shows the visibility for CASCADE at high, medium, and low traffic densities. Consider that the optimal density for high visibility is to have exactly one vehicle positioned every 300 m (*i.e.*, at the boundaries). Thus, with low density, there is less of a chance that re-broadcasted aggregated frames experience collisions, so they are able to travel farther. As the density decreases, the chances of having a single vehicle at the boundary increase. Since p-IVG dissemination used in CASCADE only deals with nearby vehicles re-broadcasting frames at the same time, there may be other transmissions (such as the initial broadcast of primary or aggregated frames and re-broadcasts from vehicles in range, but not nearby) that may collide with the transmission.

VI. Conclusion and Future Work

We have presented CASCADE, a technique for accurate aggregation of vehicle data. The local view presents data gathered from primary records, which are sent in signed frames containing a vehicle's position information. The local view is grouped into clusters, which are then used to compact and aggregate the local view data. Aggregated data from other vehicles can be used to extend a vehicle's view past its 1.5 km local view. Since vehicles' positions and speeds are represented as differences from the cluster data rather than combined with other vehicles' data, the accuracy of

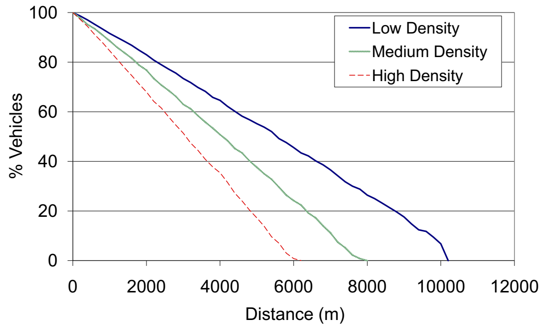

Fig. 14: Visibility with CASCADE

the aggregated data in our system is very high. We have shown through analysis and simulation that CASCADE makes efficient use of the wireless channel while providing each vehicle with data that is highly accurate, represents a large area in front of the vehicle, and can be combined with aggregated data from other vehicles to further extend the covered area.

In future work, we plan to develop algorithms for increasing security using the data in the extended frames to detect and isolate vehicles that lie about their position or speed or that intentionally mis-aggregate data. Currently, our analysis and compression methods are based on the assumption of a four-lane highway. We plan to develop methods that allow for various sized highways and that take advantage of vehicles traveling in the opposite direction for dissemination and corroboration of vehicle data.

VII. Acknowledgments

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

References

[1] US DoT, "Incident Management: Detection, Verification, and Traffic Management," Sep. 1998.

[2] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode, "TrafficView: Traffic data dissemination using car-to-car communication," *ACM Mobile Computing and Communications Review (MC2R), Special Issue on Mobile Data Management*, vol. 8, no. 3, pp. 6–19, Jul. 2004.

[3] 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.

[4] 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.

[5] C. Lochert, B. Scheuermann, C. Wewetzer, A. Luebke, and M. Mauve, "Data aggregation and roadside unit placement for a VANET traffic information system," in *Proceedings of ACM VANET*, San Francisco, CA, Sep. 2008, pp. 58–65.

[6] B. Yu, J. Gong, and C.-Z. Xu, "Catch-up: A data aggregation scheme for VANETs," in *Proceedings of ACM VANET*, San Francisco, CA, Sep. 2008, pp. 49–57.

[7] H. Saleet and O. Basir, "Location-based message aggregation in vehicular ad hoc networks," in *Proceedings of IEEE AutoNet*, Washington, DC, Nov. 2007, pp. 1–7.

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

[9] 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.

[10] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TAG: a tiny aggregation service for ad-hoc sensor networks," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, pp. 131–146, 2002.

[11] B. Przydatek, D. Song, and A. Perrig, "SIA: Secure information aggregation in sensor networks," in *Proceedings of ACM SenSys*, Los Angeles, CA, Nov. 2003, pp. 255–265.

[12] Y. Yang, X. Wang, S. Zhu, and G. Cao, "SDAP: A secure hop-by-hop data aggregation protocol for sensor networks," in *Proceedings of ACM Mobihoc*, Florence, Italy, May 2006, pp. 356–367.

[13] C. Lochert, B. Scheuermann, and M. Mauve, "Probabilistic aggregation for data dissemination in VANETs," in *Proceedings of ACM VANET*, Montreal, Canada, Sep. 2007, pp. 1–8.

[14] T. ElBatt, S. Goel, G. Holland, H. Krishnan, and J. Parikh, "Cooperative collision warning using dedicated short range wireless communication," in *Proceedings of ACM VANET*, Los Angeles, CA, Sep. 2006, pp. 1–9.

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

[16] US DoT, "Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems," ASTM E2213-03, Aug. 2003.

[17] US DoT and IEEE, "IEEE 1609 - Family of Standards for Wireless Access in Vehicular Environments (WAVE)," Jan. 2006.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] US DoT, National Highway Traffic Safety Administration, "Identify Intelligent Vehicle Safety Applications Enabled by DSRC," DOT HS 809 859, Vehicle Safety Communications Project Task 3 Final Report, Mar. 2005.

[23] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *International Journal of Information Security*, vol. 1, no. 1, pp. 36–63, Aug. 2001.

[24] 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.

[25] P. Deutsch, "DEFLATE Compressed Data Format Specification version 1.3," RFC 1951, May 1996. [Online]. Available: http://www.ietf.org/rfc/rfc1951.txt

[26] K. Ibrahim and M. C. Weigle, "ASH: Application-aware SWANS with highway mobility," in *Proceedings of IEEE MOVE*, Phoenix, AZ, Apr. 2008, pp. 1–6.

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

[28] R. Barr, Z. Haas, 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.

[29] F. Kargl and E. Schoch, "Simulation of MANETs: A qualitative comparison between JiST/SWANS and ns-2," in *Proceedings of the International Workshop on System Evaluation for Mobile Platforms (MobiEval)*, San Juan, Puerto Rico, 2007, pp. 41–46.

[30] 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.

[31] 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.

[32] 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.

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