Hierarchical Aggregation for Delay-Sensitive Vehicular Sensing

Jagruti Sahoo[¥], Soumaya Cherkaoui[€], Abdelhakim Hafid [£]
[¥]Concordia University, Montreal, Canada, [€]University of Sherbrooke, Sherbrooke, Canada
[£] University of Montreal, Montreal, Canada
[¥]jagrutiss@gmail.com, [€]Soumaya.Cherkaoui@USherbrooke.ca, [£]ahafid@iro.umontreal.ca

***Abstract*— Vehicular sensing has gained significant attention in recent years, thanks to its enormous benefits to many entities including traffic management centers, forensic authorities and air pollution control units. To reduce redundancy and improve the content, the collected data must be aggregated. Delay-sensitive sensing applications require the aggregated data be collected with a certain delay. In this paper, we propose a hierarchical aggregation scheme for image sensing in vehicular networks. The hierarchy is dynamically updated based on observed network conditions. In particular, partition and merge operations are performed on the hierarchy to satisfy the delay requirement. The simulation results show that the proposed scheme outperforms existing scheme in terms of efficient data collection and redundancy elimination.**

Keywords*— *Aggregation; road-side unit; vehicular sensing.

I. INTRODUCTION

Vehicular sensing has emerged as a powerful sensing platform because of the interactions between sensor-equipped vehicles and the physical world. Indeed, vehicles can accommodate a wide variety of sensors such as GPS, still/video cameras, accelerometers and pollution detectors; thus, vehicular sensing offers tremendous opportunity to visualize the dynamics of the environment at any instant, or over a period of time. Vehicular sensing in urban environments [2] has many dimensions namely capturing surveillance videos of streets, sensing road congestion, discovering available parking spaces and measuring the concentration of carbon dioxide (CO_2). Image/video sensing [5][11] is one application that has gained popularity in vehicular networks.

Vehicles that have still/video cameras can capture snapshots or record short videos of streets. The real-time images can be used by police to track criminal activities on roads. Besides, image/video sensing applications are of paramount importance in intelligent transportation systems especially in advanced driver assistance systems [12][13] which enhance the visual perception of drivers. Images provide precise information to determine road traffic parameters such as vehicle quantity, vehicle speed, vehicle density and flow rates. Since multiple vehicles travel through a same road segment, image redundancy is inevitable both in time and space. Thus, images must be processed to eliminate redundant images; this kind of processing is termed data aggregation.

In this paper, we consider that sensed images are aggregated in vehicular network and sent to a server through Dedicated Short Range Communications (DSRC) road-side units (RSUs). Due to their high installation costs, RSUs are expected to be deployed only at major intersections and/or busy roads. In the absence of a hierarchy, sensed images are directly sent to an aggregator (i.e. a vehicle that performs aggregation) within the coverage of RSUs. However, with an increase in the number of vehicles, severe communication contention results in excessive delay because of common path (usually the shortest path) used to deliver images to RSUs. As a consequence, a destination RSU may receive a small fraction of the images being sensed within a specified time period. To reduce contention and allow collection of sensed images within the specified time period, we propose a hierarchical aggregation scheme in which sensed images are aggregated in small regions. We periodically observe the delay of image collection and update the hierarchy in order to satisfy the delay requirement.

The rest of the paper is organized as follows. Section 2 presents related work. Section 3 describes the proposed scheme. Section 4 evaluates the scheme through simulations. Finally, Section 5 concludes the paper and presents future work.

II. RELATED WORKS

CarTel [3] is a versatile sensing platform that offers services such as traffic monitoring, environmental monitoring, automotive diagnostics and geo-imaging. A CarTel node can act as “data mule” and the sensed data is delivered in best-effort manner. Because of its delay-tolerant nature, CarTel is less efficient for real-time acquisition of sensed data.

Some contributions [4][7][8] consider deployment of RSUs to collect traffic data sensed by vehicles. RSUs can send the received data to a server for further analysis. The sensing task is initiated either on-demand or is performed proactively. In [7], a few RSUs are defined as task organizers (TOs). TOs are deployed at various positions along the road. To collect traffic data (e.g. traffic density, mean speed and travel times), TO broadcasts a message containing its location; vehicles that receive the message trigger the sensing task and send the sensed data back to TO. This scheme is designed for a single road segment and moreover it assumes the presence of DSRC road-side units along the roads. However, in reality it is infeasible to deploy RSUs along all or a majority of road segments in a city. To reduce the amount of data transmitted

978-1-4799-5344-8/15/$31.00 ©2015 IEEE

to RSUs, the authors in [4] propose an in-network aggregation scheme. An efficient in-network aggregation creates fewer content-rich reports out of many sensor readings.

In [4], a cluster-based architecture is proposed for the collection of aggregated traffic data at RSUs owned by Wimax or 2.5/3G providers. A road is divided into a number of small segments; a cluster head (i.e. a vehicle) is selected for each segment. The cluster head of the farthest segment, with respect to RSU, aggregates the information sensed in its segment and sends it to the cluster head of next segment. Finally, the cluster head in the closest segment (to RSU) sends the aggregated data to RSU. Using this methods, it is not possible to collect sensed data in a complex environment (e.g. entire city), as it would involve a huge cost to deploy RSUs on all road segments.

The DB-VDG protocol proposed in [15] is closely related to our work in the way that it addresses delay-bound data collection in urban vehicular sensor networks. In this protocol, data is collected within a circular region centered at the base station. Vehicles transmit sensed data to a base station using one of the two approaches: data muling and next hop forwarding. A vehicle adopts data muling, if the time to reach the base station is smaller than the remaining query life time. Otherwise, the vehicle selects a neighbor that has lowest aggregation level (amount of data carried) as the next hop. The limitation of DB-VDG is that network dynamics is not considered for deciding aggregation operations. As a result, the potential of aggregation is not harnessed completely. Moreover, for query with short life-time, aggregation is performed on data received from very few sources and as a result, redundant packets are transmitted over several hops in the networks.

Several approaches [1][5][6][9] are proposed for distributing sensed data in vehicular networks. In [1], [5] and [9], proactive approach is used meaning that vehicles broadcast sensed information periodically. Information received from various vehicles are aggregated and broadcasted to other vehicles in the network. Proactive approach needs strict collaboration among vehicles in order to avoid the eventual packet loss. Packet loss is eliminated in [9] by using the Space Division Multiple Access (SDMA) technique. However, the authors only focus on traffic information. On the other hand, reactive approach is used in [6], wherein vehicles obtain required sensed information by disseminating queries. The drawback of reactive approach is that it is difficult to obtain data of a past sensed event and the temporal scope of sensed data is limited to a few seconds rather than hours, days or weeks.

III. PROPOSED SCHEME

In this paper, we propose a dynamic hierarchical aggregation (DHA) scheme with a focus on image sensing applications in urban scenarios. We consider that images captured through cameras installed on vehicles are aggregated in the network and uploaded to a central server through a DSRC RSU for further processing.

A. *Network Model*

We consider that sensing is performed in a given region, called sensing region. A sensing region is defined as a set of geographically connected road segments and contains one RSU. We consider the entire network as the sensing region which is a connected graph $G = (V, E)$, where each vertex $v \in V$ denotes an intersection and each edge $e \in E$ denotes a road segment. If RSU is located on an edge (u, v), the edge is divided into two edges (u, k) and (k, v), where k denotes the location of RSU. On the other hand, when RSU is located at an intersection, it can be referred using the vertex that denotes that intersection. Similarly, when vehicles communicate, their positions can be considered as vertices.

B. *Basic Idea*

In our scheme, all vehicles are allowed to sense images; hence redundancy is inevitable. Further, images captured on different streets bear similarity to some extent. Aggregation is performed to eliminate the redundant images. We consider that aggregation is performed by vehicle(s) termed as aggregator(s). For data (i.e. sensed/aggregated image) dissemination, vehicles use 802.11 based directional broadcast. A list of intersections is provided in the header of the broadcast packet; the list is essentially a shortest hop-count path. Hop-count is considered as the cost of an edge and then Dijkstra's algorithm [10] is executed to find the shortest path. A packet is broadcasted following the list until it reaches the road segment containing the aggregator.

If sensed images are sent directly to one aggregator (i.e. vehicle located within the coverage range of RSU), then packets from multiple vehicles are likely to share either the same path or a portion of it along their way to the aggregator. When the number of vehicles increases in the sensing region, this will result in heavy contention among vehicles attempting to transmit or forward images. Moreover, in a highly dynamic vehicular scenario, the vehicle density is likely to fluctuate very rapidly. Consequently, the image sensing application will experience serious performance degradation in terms of its inability to produce the required information in real-time. We overcome this problem by using a hierarchy for data collection. The hierarchy is rooted at the aggregator located within the coverage range of RSU. Each node in the hierarchy is either an aggregator or a set of non-aggregators. A non-aggregator denotes a vehicle that generates sensed image. An aggregator can have a set of non-aggregators and other aggregators as its child nodes. Thus, an aggregator can receive sensed images directly from vehicles as well as aggregated images from other aggregators. The aggregator then performs aggregation using both sensed images and aggregated images and sends the aggregated images to its parent node in the hierarchy. Final aggregation is performed at the root.

C. *Hierarchical Aggregation*

We consider the image sensing application as delay-sensitive. It implies that the performance of the sensing application is satisfactory when sensed images (either aggregated or original) are received at the root aggregator within a predefined delay threshold. The notion of delay threshold stems from the necessity to reconstruct a scene that spans one or more road segments in real-time. Let us denote T_{col} as the duration of the image collection interval. T_{col} consists of delay threshold T_{img}, and a small guard time T_w.

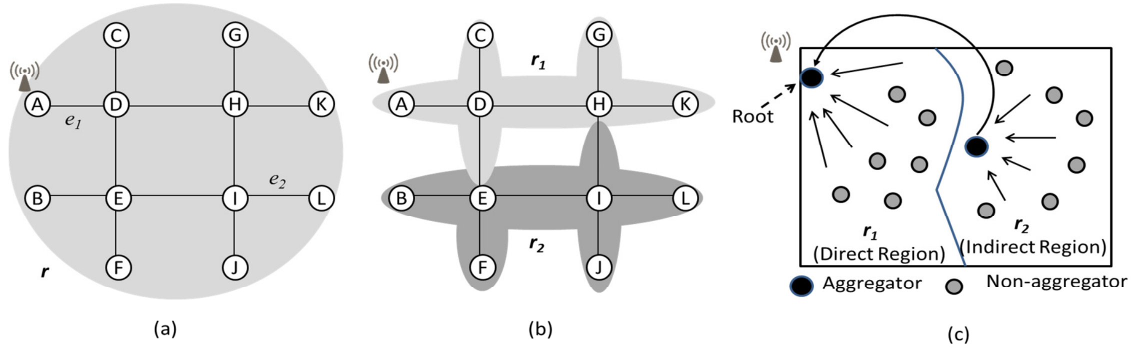

Fig. 1. (a) Initial region r, (b) Output regions r_1 and r_2 by applying PA on region r (c) Image Collection at root.

The guard time is necessary to allow reception of images even when the delay threshold is not satisfied. At the beginning, the hierarchy consists of only root aggregator. We allow few image collection intervals to pass in order to obtain certain statistics (i.e., vehicle count and vehicle arrival rates) described later which would be used to update the hierarchy. Because of dynamic topology in vehicular networks, the ratio of number of packets received at the root aggregator within the delay threshold to the total number of packets originated in the network may vary from one interval to another. Thus, after the required statistics are collected, the root aggregator starts to determine whether the hierarchy satisfies the delay requirement and creates new hierarchy by applying appropriate update operation.

After the delay threshold T_{img} is elapsed, root aggregator determines if the sensed/aggregated images received at the root aggregator includes images from all vehicles (each vehicle generates one sensed image) in the network. Based on the above outcome, the root aggregator decides whether to employ the update operation on the hierarchy. We employ two types of update operation: *partition* operation and *merge* operation. Partition operation partitions a given region r into two smaller regions, called *direct region* and *indirect region*. Vehicles in the direct region send sensed images directly to the aggregator of region r; whereas a new aggregator is selected for the indirect region. Vehicles in the indirect region send sensed images to the new aggregator which after aggregation sends aggregated images to the aggregator of region r.

Merge operation on the other hand reunites already existing direct region and indirect region back into their parent region r. The information regarding update is provided by root in the beginning of each image collection interval. If the hierarchy consists of only root, then partition is the only possible update operation which creates a child aggregator of the root aggregator. Because of this child aggregator, data size is reduced further and hence smaller delay is incurred at the root aggregator. However, when the hierarchy has more than one aggregator, it is dynamically updated by employing either partition or merge operation whichever incurs lower delay at the root aggregator.

If the update involves partition operation on region r, the aggregator of region r broadcast partition information (i.e. segment IDs of road segments that constitute the resultant direct region and indirect region). Let r_1 and r_2 denote direct region and indirect region of region r respectively. Non-aggregators in direct region continue to send sensed images to aggregator of region r. However, to select new aggregator in the indirect region, the aggregator of region r broadcast new aggregator location and corresponding segment ID in the indirect region. Basically, the location that results in fewer hops to the next aggregator is used to select the new aggregator. We define a small area (e.g. 300m) around this location. Vehicles located in this area exchange short messages with each other to select one of them who will stay longest time in this area as the aggregator. Once, a new aggregator is selected, it broadcasts a message to vehicles in the indirect region to start sending sensed images. If the update indicates a merge operation of a direct region r_1 and indirect region r_2 to parent region r, the aggregator of r_2 broadcast the position of aggregator of r to inform vehicles in region r_2 to send sensed images directly to aggregator of r.

In the start of each image collection interval, image collection is initiated by root aggregator by broadcasting messages. In particular, the root aggregator broadcasts its position to its non-aggregators. In addition, root aggregator also communicates with all other aggregators in the hierarchy. Each of the aggregators, after receiving message from root aggregator, broadcasts another message containing its position in to its non-aggregators.

In the following sections, we discuss aggregation factors, vehicle count prediction and dynamic update of hierarchy.

1) Aggregation Factors: Aggregation factor denotes the ratio of number of non-redundant images produced after aggregation to the total number of images. It depends on the image processing techniques to eliminate redundancy among images. For eliminating redundant images captured along a road segment, we introduce a relatively simple scheme to determine the aggregation factor, say ρ. On the other hand, to eliminate redundancy among images that belong to different road segments, we use a constant value of aggregation factor, say σ for sake of simplicity. ρ is computed as follows. We consider that position of camera is same as the position of the vehicle on which the camera is installed. Each image contains scene of a square with side X meter. Given a road segment, we first determine the minimum vehicle count, denoted by d_1, at which the entire road segment is covered and there would be no overlap among images captured by vehicles. For a road segment of length L, d_1 is computed as L/X. Then, given a vehicle count, say d_2, of the road segment, aggregation factor is given as follows:

$$\rho = 1 - \frac{d_1}{d_2} \tag{1}$$

2) Vehicle Count Prediction: In order to decide the update operation, it is necessary to determine the vehicle count of all road segments. Vehicle counts are computed by the aggregators and sent to the root aggregator. In particular, an aggregator computes the vehicle count of all road segments under its control. We propose a method that uses vehicle's kinematics profile in the current image collection interval and moving average of vehicle arrival rate to predict vehicle count in the next image collection interval. The kinematic profile of a vehicle includes its position, speed, acceleration and direction. If $n(t)$ denotes the vehicle count of a road segment at time t, vehicle count at time $t+T_{img}$ can be expressed as follows:

$$n(t+T_{img}) = n(t) + \lambda_a(t, t+T_{img}) + \lambda_d(t, t+T_{img}) \tag{2}$$

where $\lambda_a(t,t+T_{img})$ represents the number of vehicles that enter the road segment in the time interval $(t, t+T_{img})$ and $\lambda_d(t, t+T_{img})$ represents the number of vehicles that depart the road segment in the time interval $(t, t+T_{img})$. Note that, vehicles send their kinematics information along with their sensed images to the aggregator in charge. Vehicle count $n(t)$ is obtained from the received position of vehicles. Also, using speed and acceleration values, vehicles that will no longer stay in the same road segment, after a duration of T_{img}, can be identified and hence $\lambda_d(t, t+T_{img})$ can be obtained. Aggregators observe vehicle arrivals in each image collection interval and compute $\lambda_a(t, t+T_{img})$ as the moving average of K most recent observations. After computing the components of (2), aggregators then obtain the vehicle count for next image collection interval. Since only vehicle counts are sent to the root aggregator, this method does not incur extra overhead while still provides crucial information.

3) Dynamic Update: In each image collection interval, the hierarchy is updated, if required. The update is achieved by performing either partition operation or merge operation on nodes of the hierarchy. While merge operation is straightforward as accomplished by taking union of edge sets and vertex sets; partition operation needs an algorithm to distribute edges/vertices into two different regions. The pseudo code of the partition algorithm (PA) is given in Table-I.

PA is used to divide a given region r into two smaller regions r_1(direct region) and r_2 (indirect region). Region r is given by a connected graph G'. PA is based on breadth-first traversal of edges. The objective is to traverse edges of G' in order to produce two connected sub-graphs G_1 and G_2 as the output regions. First, we select the edge on which aggregator of r is located as the input edge e_1 for the direct region. Then, edge e_2 which is farthest from e_1 is selected as input edge for the indirect region. The traversal begins at the selected edges. E_1 and E_2 denote the set of edges for two output regions. During the execution of the first while loop in PA, edges are en-queued and de-queued. After an edge is de-queued, only one of its neighboring edges is visited in each iteration of the while loop. The de-queued edge is added to the edge set of the output region only when all of its neighboring edges are visited.

Fig. 1(a) shows region r where edges e_1 and e_2 are selected as the input edges. Fig. 1 (b) shows region r_1 and region r_2 produced as a result of execution of PA on region r. Fig. 1(c) shows collection of images at the root aggregator. As shown in Fig. 1 (c), in r_1, the non-aggregators send sensed images directly to the root aggregator; whereas in r_2, the sensed images are aggregated at an intermediate aggregator before they are sent to the root aggregator.

Table-1: Partition Algorithm (PA)

Input: Connected Graph $G'=<V',E'>$ (Region r)
Initialization: Select $e_1 \in E'$ s. t. e_1 is the edge on which aggregator of region r is located.
Select $e_2 \in E'$ s. t. e_2 is farthest from e_1
$G_1 =< V_1, E_1 >, V_1 = E_1 = \phi$ (Region r_1)
$G_1 =< V_2, E_2 >, V_2 = E_2 = \phi$ (Region r_2)
$Q_i = \phi$ is a queue defined for region r_i
For an edge w, $N(w) = \{z \in E' : w$ and z share a common ver tex$\}$
$\forall e \in E' - \{e_1, e_2\}$, *visited* $[e]$ = false
visited $[e_1]$ = true
visited $[e_2]$ = true

while ($\exists e \in E'$ for which *visited* $[e]$!= true)
  for $i = 1$ to 2
    if $\exists e \in N(e_i)$ and *visited* $[e]$!= true *then*
      Enqueue (Q_i, e)
      visited $[e]$ = true
    end if
    if visited $[e]$ = *true* $\forall e \in N(e_i)$ *then*
      $E_i = E_i \cup \{e_i\}$
      $e_i = Dqueue(Q_i)$
    end if
  end for
end while
for $i = 1$ to 2
  while (Q_i is nonempty)
    $e_i = Dqueue(Q_i)$
    $E_i = E_i \cup \{e_i\}$
  end while
  for each edge $e = \{u, v\} \in E_i$
    $V_i = V_i \cup \{u, v\}$
  end for
end for

Given a hierarchy, when partition operation is applied, the new hierarchy is formed as follows. The direct region of root aggregator is partitioned into two regions: direct region and indirect region. For new direct region, the branch ends with a non-aggregator node. For new indirect region, the branch contains a new aggregator node as the child of the root and a non-aggregator node as the child of the new aggregator node. This procedure is repeated for all aggregators in the indirect region of root. Unlike partition operation, in merge operation, the direct region and indirect region are merged and hence the branches are replaced by one branch that ends with a non-aggregator node.

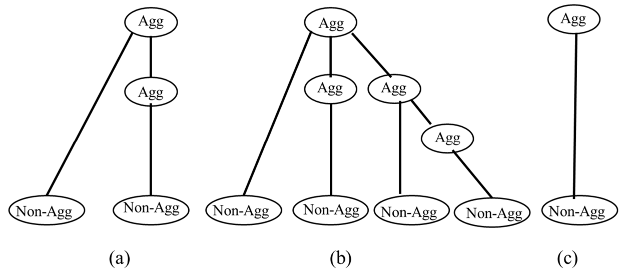

Fig. 2. (a) Original Hierarchy, (b) Hierarchy after partition operation, (c) Hierarchy after merge operation.

This procedure is repeated for each pair of direct region and indirect region in the hierarchy. Fig. 2. (b) and Fig. 2(c) show the new hierarchy after applying partition operation and merge operation respectively on the hierarchy shown in Fig. 2. (a).

Partition operation is needed when number of vehicles increases in each region. By partitioning, a region is divided into two smaller regions and as a result contention decreases which in turn decreases the delay of transmitting sensed images to root aggregator. Similarly, merge operation is needed when the delay incurred in sending sensed images through a child aggregator is greater than sending sensed images directly to the parent aggregator. In order to identify the appropriate update operation, we quantify the delay at the root aggregator. Since contention is proportional to the number of packet transmissions; we consider latter as the quantifier of delay. We compute the total number of transmissions in the hierarchy produced by each update operation. Then, the operation that results in lowest number of transmissions in the hierarchy is selected. The number of transmissions in a given hierarchy is expressed as follows:

$$N_{tr} = \sum_{i=1}^{N_{n-agg}} \sum_{e \in E(r_i)} h_{i,e} * n_{i,e} + \sum_{j=1}^{N_{agg}} H_j * s_j \quad (3)$$

where $N_{n\text{-}agg}$ and N_{agg} denote number of non-aggregator nodes and aggregator nodes in the hierarchy respectively. $h_{i,e}$ and $n_{i,e}$ denote hop-count from edge e in region r_i to aggregator of region r_i and vehicle count on edge e (i.e. road segment) respectively. Since each vehicle sends exactly one packet; vehicle count of an edge represents the total number of packets to be originated on that edge. H_j and s_j denote hop-count from aggregator j to its parent aggregator and number of packets sent by aggregator j respectively. To compute N_{tr}, the first expression in (3) requires predicted vehicle count information; the second expression requires predicted number of packets after aggregation. Aggregators send the above information while sending the aggregated packets and hence enable the root aggregator to compute N_{tr} and identify the update operation. Once the vehicle count is predicted (as described in Section-III. C. 2), the aggregation factor ρ can be obtained easily (as described in Section-III. C. 1) and so is the number of packets that will be generated in a road segment. Then, using ρ, the predicted number of packets after aggregation can also be obtained.

IV. Performance Evaluation

A. *Simulation Setup*

In this section, we investigate the performance of the proposed scheme DHA using ns-2 simulator. We used version ns-2.33, which is modelled to provide MAC and PHY layer specifications of 802.11p [14]. We compare DHA with DB-VDG protocol [15]. To illustrate the limitation of a static hierarchy for data collection, we consider a baseline scheme SHA (Static Hierarchical Aggregation), in which the hierarchy consists of only one intermediate aggregator. σ *is* set to 0.1 and X is set to 40m. The simulation scenario consists of 800 x 800 m^2 grid road network with 25 intersections. The length of each road segment is equal to 200m.

TABLE-II.
Simulation Parameters

Parameter	Value
Image Collection Interval	5 Sec
Channel Propagation	TwoRayGround
Data Rate	6 Mbps
Transmission Range(Vehicle and RSU)	300m
Simulation Time	200 sec

RSU is placed at the center of the network. Velocity of vehicles varies between 15m/s and 25m/s. We consider an image sensing application for the evaluation; it is simulated by generating a packet of size 1000 Bytes. Each vehicle sends the sensed image periodically in each image collection interval. Other simulation parameters are listed in TABLE-II.

B. *Performance Metrics*

We use the following metrics for evaluating the proposed scheme.

1) *Delivery Efficiency:* It is defined as the ratio of number of packets (in original form or in aggregated form) received at the root aggregator within the image collection interval to the total number of packets originated in the network.

2) *Aggregation ratio:* It is defined as the number of packets produced after aggregation to the total number of packets originated in the network.

C. *Results and Discussions*

Fig. 3 shows delivery efficiency with respect to the total number of vehicles. We observe a sharp degradation in delivery efficiency of SHA with increase in number of vehicles. This is due to the fact that, aggregation is performed only once. On the other hand, DB-VDG outperforms SHA, as the former approach involves in-network aggregation. The proposed protocol DHA attains the highest value of delivery efficiency. The rationale lies in the fact that DHA adapts to varying network condition and updates the hierarchy accordingly. The reason behind lower delivery efficiency in case of DB-VDG is the criteria used for next hop selection. DB-VDG chooses a vehicle which has lowest aggregation level (i.e. number of data packets carried by the vehicle). Due to this criteria the inherent redundancy among images captured by vehicles in close proximity is not eliminated, which leads to higher contention and more delay. On the other hand, we eliminate image redundancy by allowing a vehicle to perform aggregation on all images captured by neighbors.

Fig. 4 shows the aggregation ratio with respect to total number vehicles in the network. Lower value of aggregation ratio shows low bandwidth usage by a scheme. We notice that DHA has the lowest value of aggregation ratio for all values of number of vehicles. SHA has the worst performance by attaining an aggregation ratio of 0.73 at the highest number of vehicles. The aggregation ratio in case of DB-VDG lies in between SHA and DHA. With increase in number of vehicles, images captured along a road segment are more likely to overlap compared to when there are fewer vehicles on the road segment. However, in DB-VDG, vehicles select next-hop forwarders independently. Thus, there can be several vehicles that perform aggregation on a very small number of images and again forward the aggregated image to next hop forwarders. As a result, number of packets is not reduced. In contrast, in our proposed scheme, all vehicles within a small region send their images to a single aggregator which then performs aggregation; leading to a substantial elimination of redundant images.

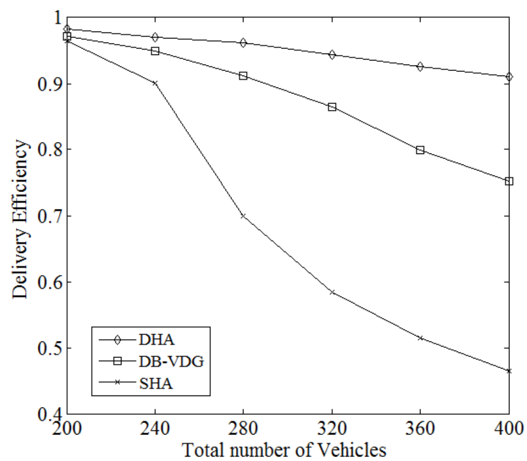

Fig. 3. Delivery Efficiency

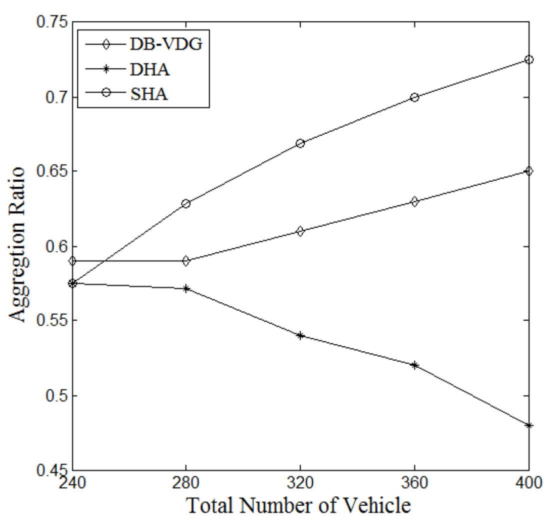

Fig. 4. Aggregation Ratio

V. CONCLUSION AND FUTURE WORK

Vehicular sensing applications become more popular as vehicles embed an increasing number of sophisticated sensors. Information such as images, when collected over time from a large number of vehicles can congest the network and may fail to arrive in real time to RSU. In this paper, we address this problem by proposing a scheme for sensed data (such as images) aggregation in a hierarchical way in order to decrease the volume of data transferred from the road and also to decrease the delay of delivering sensory information.

In future, we plan to evaluate the proposed scheme for other sensing applications such as traffic information collection and parking space discovery.

REFERENCES

[1] Y. Dieudonn, B.Ducourthial, and S.M. Senouci, "COL: a Data Collection Protocl for VANET", in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, pp.711-716, 3-7 June 2012.

[2] U. Lee and M. Gerla, "A survey of urban vehicular sensing platforms," Computer Networks, vol. 54, no. 4, pp. 527–544, 2010.

[3] B. Hull et al., "CarTel: A Distributed Mobile Sensor Computing System," in *Proc. ACM SenSys*, pp. 30-39, 2006.

[4] I. Salhi, M. O. Cherif, S. M. Senouci, "A New Architecture for Data Collection in Vehicular Networks," In *Proc. IEEE ICC*, pp.1,6, 14-18 June 2009.

[5] U. Lee, E. Magistretti, M. Gerla, P. Bellavista and A. Corradi," Dissemination and harvesting of urban data using vehicular sensor platforms", *IEEE Transaction on Vehicular Technology,* vol.58, no.2, pp.882,901, Feb. 2009.

[6] M. D. Dikaiakos, A. Florides, T. Nadeem, L. Iftode, "Location-Aware Services over Vehicular Ad-Hoc Networks using Car-to-Car Communication," *IEEE Journal on Selected Areas in Communications* , vol.25, no.8, pp.1590-1602, Oct. 2007.

[7] H. Arbabi and M.C. Weigle, "Using Vehicular Networks to Collect \Common Traffic Data", in *Proc. ACM VANET*, Beijing, China, 2009, pp. 117-118.

[8] A. Miloslavov, M. Veeraraghavan, "Sensor Data Fusion Algorithms for Vehicular Cyber-Physical Systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, pp.1762-1774, Sept. 2012.

[9] W. Chang, H.T. Lin, B. Chen, "TrafficGather: An Efficient and Scalable Data Collection Protocol for Vehicular Ad Hoc Networks," in *Proc. IEEE CCNC*, pp.365-369, 10-12 Jan. 2008.

[10] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, "Section 24.3: Dijkstra's algorithm".Introduction to Algorithms (Second ed.). MIT Press and McGraw-Hill. 2001, pp.595–601.

[11] R. Bruno, M. Nurchis, "Efficient data collection for multimedia vehicular sensing platforms ," *Pervasive and Mobile Computing*, vol. 16, Part A, pp 78-95, January 2015.

[12] M. A. Bauer, K. Charbonneau, S. S. Beauchemin, "V2Eye: Enhancement of visual perception from V2V communication," in *Proc. IEEE CCNC*, pp.818,819, 9-12 Jan. 2011

[13] P. Gomes, C. Olaverri-Monreal, M. Ferreira, "Making Vehicles Transparent Through V2V Video Streaming," *IEEE Transactions on Intelligent Transportation Systems*,vol.13, no.2, pp.930,938, June 2012.

[14] C. Campolo, H. A. Cozzetti, A. Molinaro, R. M. Scopigno, "Overhauling ns-2 PHY/MAC simulations for IEEE 802.11p/WAVE vehicular networks," in *Proc. IEEE ICC*, pp.7167,7171, 10-15 June 2012.

[15] C. E. Palazzi, F. Pezzoni, and P. M. Ruiz, "Delay-bounded data gathering in urban vehicular sensor networks," *Pervasive Mobile Comput*., vol. 8, no. 2, pp. 180–193, Apr. 2012.