Traffic Density Estimation Protocol Using Vehicular Networks

Adnan Noor Mian[1], Ishrat Fatima[1], and Roberto Beraldi[2]

[1] National University of Computer and Emerging Sciences,
Block-B, Faisal Town, Lahore, Pakistan
{adnan.noor,ishrat.fatima}@nu.edu.pk
[2] DIS, Università di Roma *"La Sapienza"*
via Ariosto, 25, Roma, 00185, Italy
beraldi@dis.uniroma1.it

Abstract. Traffic density estimates on maps not only assist drivers to decide routes that are time and fuel economical due to less congestions and but also help in preventing accidents that may occur due to the lack of not being able to see far ahead. In this paper, we propose a protocol that exploit vehicle-to-vehicle ad hoc communication for the estimation of vehicular density and the amount of congestion on roads. The protocol forms cluster heads by a voting algorithm. These cluster heads aggregate density information and spread it to the network via few selected forwarding vehicles. The protocol does not assume all vehicles to be equipped with Global Positioning System. We analytically study the cluster head formation part of the protocol and then simulate the proposed protocol using network simulator NS2 to understand different characteristics of the protocol.

Keywords: Vehicular networks, traffic congestion, wireless ad hoc networks.

1 Introduction

Congestion estimation can be used in providing the efficient route information to the drivers. Generally two approaches can be employed for getting traffic congestion information using vehicular network. The first approach uses road side infrastructure in addition to the vehicles and the second only exploit the vehicles on the road and uses only vehicle-to-Vehicle (V2V) communication for estimating traffic congestion. In this paper we focus on the second approach.

There are many protocols available for collecting road traffic information using V2V communication, however, most of them work by generating large amount of network traffic [6][12][9]. Huge network traffic adversely effects the performance of the protocol in many ways. Firstly, large network traffic results in collisions of packets which may eventually lead to loss of information[17], [5], such as dropping of critical messages generated by emergency and safety services [2]. Secondly, security is another reason to avoid higher network traffic generation

K. Zheng, M. Li, and H. Jiang (Eds.): MOBIQUITOUS 2012, LNICST 120, pp. 1–12, 2013.

in case of vehicular networks. Each packet in the network is digitally signed and checked using its digital signature to ensure that it was received from legitimate sender. This security clearance takes some milliseconds for each packet thus introducing large delays in the processing which in turn affects processing real time capabilities of vehicular networks [7]. Lastly, due to large network traffic the CSMA/CA mechanism of IEEE 802.11p, which is now a widely adapted standard for vehicular communication, a node has to wait before being able to send its information. This becomes a serious issue when vehicular density is high, resulting in large latencies in broadcasts.

Moreover the existing road traffic estimation protocols depend on the GPS readings and assume that all the vehicles have GPS installed. Their performance will be drastically affected in scenarios where the number of GPS equipped vehicles are less. For example, mostly in the developing countries small number of vehicles may have GPS installed. Even if the GPS is installed, the GPS outages can occur in different areas like tunnels and hilly areas[4]. There are techniques like dead reckoning [13] and infrastructure assisted localization [10] that aim to find the position during GPS outages but these solutions are computationally intensive. Such reasons motivates for a protocol that can work fairly even when the GPS availability is not 100% in the vehicular network.

In this paper, we present a Traffic Density Estimation Protocol (TDEP) which utilizes minimum bandwidth by generating minimum network traffic. The protocol works by electing cluster heads. These cluster heads gather the road traffic information for a certain period of time and then broadcast this information. Only the vehicles decided by the cluster heads rebroadcast this information. The protocol does not require every vehicle to have GPS. The vehicles that do not have GPS can only transmit their ids to cluster heads to inform about their presence. We show by our simulations that the proposed protocol works well even if there are only 50% of the vehicles have GPS. We not only do simulations to analyze the behavior of the protocol under different traffic scenarios but also provide a mathematical analysis of the cluster head formation part of the protocol, which forms the core of the protocol.

2 Related Work

Our approach is based on cluster head formation but is different from the Connected Dominating Set (CDS) approach[3]. In this approach the vehicles in the virtual backbone of CDS may become congested with network traffic and may drop packets whereas in the proposed protocol each cluster head will select few vehicles for forwarding packets thus distributing the load. Also it incurs no cost for maintaining the virtual backbone, which is otherwise expensive in such scenarios [15].

Porikli and Li [14] propose an algorithm that exploit traffic videos for the estimation of congestion. In [16], Singh and Gupta assume that the vehicles in proximity contribute more for congestion. Pattara et. al. [18] exploit the cellular technology and use Cell Dwell Time (CDT) to determine the congestion. If a mobile cell has larger CDT the longer the vehicle remains in contact with one base station and the higher is the probability of congestion. Hang et. al. [8] use shockwaves to identify congestion on a road. Shockwave is produced when an unusual event e.g., an accident occurs at the road. In Padron [12] approach when a vehicle considers its speed to be smaller than a threshold, it votes for their own speed to be lower. If a certain number of vehicles around this vehicle also vote for the same, the congestion is supposed to occur and the information is then flooded to the network. The CASCADE protocol [9] aggregate data and rebroadcast the aggregated data for the purpose of developing a position map of vehicles. It divides the area in front of a vehicle into 12 clusters and aim to display the exact position of vehicle on the map. TrafficView [11] protocol display the traffic scenario to the driver by aggregating a vehicle's neighborhood information and flooding it to other vehicles. Chang et. al. [6] Trafficgather protocol forms cluster of vehicles and finds cluster heads but there is no criteria for becoming a cluster head. A node that wishes to collect information declares itself as cluster head. Furthermore Trafficgather use TDMA to avoid collision whereas TDEP use widely proposed IEEE 802.11p CSMA protocol.

The research works presented so far is different in many ways from the work proposed in this paper. We assume that each vehicle has only IEEE 802.11p protocol wireless support and do not use any other devices like directional antenna, camera or support from cellular network. The proposed protocol does not explicitly declare the presence of congestion, instead it displays approximate vehicle density road ahead to the drivers from which they can judge the amount of congestion at different places. Besides, we have done a mathematical analysis on the expected number of cluster heads, which lacks in the previously proposed protocols.

3 Proposed Protocol

The proposed protocol is given by algorithm 1. The protocol runs on each vehicle i in the network. Its different phases run asynchronously and concurrently as described below.

Cluster Head Selection. Cluster head is selected on the basis of the number of neighbors d, i.e., the number of vehicles in the transmission range. The vehicle which has more d has more chance of being selected as a cluster head. The cluster head aggregate information about its neighbor vehicles and then forward this information to other vehicles and cluster heads. The timer vehInfoTimer on each vehicle expires periodically after a specified time interval (line 1-3) to broadcasts VehInfoPkt packet. This packet contains the id of the vehicle, its GPS position if available and its no. of neighbors d. Each vehicle receiving VehInfoPkt stores the packet to its VehInfo cache (line 34-35).

The election of cluster head is done periodically, triggered by timer voteCastTimer. In this phase, each vehicle determines its d and adds this information to the VehInfo cache. The VehInfo cache is periodically searched to find the id

Algorithm 1. Protocol running on a vehicle i

```
ScheduleEvent(vehInfoTimer)
d ← number of vehicles in Location cache
bcast VehInfoPkt.i.d

ScheduleEvent(voteCastTimer)
elected_vehicle ← vehicles with max d in VehInfo Cache
if elected_vehicle > 1 then
   elected_vehicle ← random selection from elected_vehicle
   bcast VotePkt.elected_vehicle
end if

ScheduleEvent(locationUpdateTimer)
location ← ∅
if GPS is avaliable then
   location ← coordinates of vehicle i from GPS
   bcast LocationPkt.i.location
end if

ScheduleEvent(clusterHeadInfoTimer)
if clusterhead = 1 then
   d ← number of vehicles in Location cache
   Fnodes ← forwardingNodes()
   location ← coordinates of vehicle i from GPS
   bcast ClusterHeadInfoPkt.i.d.Fnodes.location
end if

ScheduleEvent(mapDisplayUpdateTimer)
for all records in ClusterHeadInfo cache do
   display point at record.location
   compute rectangular area A = 2rb / mapScale with its center positioned at location
   display record.d uniformly distributed points in A
end for

OnReceive(VehInfoPkt)
save VehInfoPkt in VehInfo cache

OnReceive(LocationPkt)
save LocationPkt in Location cache

OnReceive(VotePkt)
if VotePkt.elected_vehicle = i then
   vote ← vote + 1
end if
if vote ≥ t_f × d then
   clusterhead ← 1
else
   clusterhead ← 0
end if

onReceive(ClusterHeadInfoPkt)
save ClusterHeadInfoPkt in ClusterHeadInfo cache
if ClusterHeadInfoPkt.Fnodes = i then
   if ClusterHeadInfoPkt.ttl > 0 then
      ClusterHeadInfoPkt.ttl ← ClusterHeadInfoPkt.ttl - 1
      Fnodes ← forwardingNodes()
      bcast ClusterHeadInfoPkt.Fnodes
   else
      drop ClusterHeadInfoPkt
   end if
end if
```

of the vehicle which has a maximum value of d. If the found id is not the id of vehicle itself, then a VotePkt packet is formed which has the id having maximum d in the vote field. The packet is then broadcasted (line 5-10). When a vehicle receives a VotePkt packet, it checks if the packet has its id in the vote field. If this is the case, then the number of vote of the vehicle is incremented by one. A vehicle is decided to be the cluster head when $vote \geq t_f \times d$ where $0 \leq t_f \leq 1$ is the threshold fraction (line 40-48). This is further explained in section 4.

Density Calculation around Cluster Heads. Each vehicle periodically broadcasts a Location packet after a time interval Lt given by locationUpdateTimer. This Location Packet contains id of the vehicle and its location taken from the GPS (lines 12-17). When a vehicle receives a Location Packet, it saves this information in its Location Cache (lines 37-38). This Location Cache is maintained till the timer clusterHeadInfoTimer expires after interval Ct on the cluster head. The expiry intervals of locationUpdateTimer and clusterHeadInfoTimer follows the following condition $Ct \geq 2 \times Lt$ to refrain from redundant traffic. On expiry of clusterHeadInfoTimer timer, Location cache is traversed to calculate the number of vehicles within the range (we assume 250m) of cluster heads. The cluster head also calculates the forwarding vehicles which will be responsible for rebroadcasting the ClusterInfo packets. Cluster head broadcasts a ClusterInfo packet and initializes it with its current GPS coordinates, number of vehicles around and list of ids of forwarding vehicles (lines 19-25).

Density Estimation Propagation. When a vehicle receives a ClusterHeadInfoPkt packet, it saves the location of the cluster head and its neighbor ids into its ClusterHeadInfo Cache. It then checks the list of forwarding vehicles inside ClusterHeadInfoPkt to find out if this vehicle is the forwarding vehicle and if this is so then the packet is accepted otherwise discarded. After accepting the ClusterHeadInfoPkt, the vehicle calculates its own forwarding vehicles and replaces the list of ids of these forwarding nodes with the previous one in the ClusterHeadInfoPkt packet. The packet is then broadcasted (lines 50-60).

Map Display. The map display on each vehicle is refreshed after a certain time period. The protocol (line 27-32) identifies a rectangular area $A = r \times b$ on the map where r is the length of transmission range r and b is the breadth of road and the center of the rectangle being positioned at the cluster head. The number of neighbors of each cluster head is read from the ClusterHeadInfo cache and corresponding to each cluster head, location points are uniformly distributes points in A. We take 250m as the range for both calculating the information about cluster and then distributing its information on the map. If we increase this range then it would mean that we are aggregating information from a larger area at just one cluster head resulting in more error between the actual road scenario and the estimated map. If we decrease this range, the error again increases as we will be aggregating information of very small area at one cluster head resulting in large number of cluster heads. This will again lead to the flooding scenario, requiring too many vehicles to generate information packets and broadcast them.

Forwarding Vehicles Selection. Cluster heads and forwarding vehicles compute their forwarding vehicles for forwarding information packets to the network. The basic idea is to divide the communication range of 250m around the cluster head into equal sectors and then choose one farthest vehicle from each sector. We choose one vehicle from each sector so that we can cover most of the area around the cluster head and information can propagate in all the directions with minimum redundant packets. The number of sectors can vary. Here we have set the number of sectors equal to 4.

4 Analysis of Expected Number of Cluster Heads

We know that the number of cluster heads effect the performance of the protocol. The expected number of cluster heads, in turn depends on the value of the threshold in the protocol. In this section we shall derive an analytical expression for the expected number of cluster heads when the threshold is varied. Let us take a snap short of an urban congested scenario. In such a case vehicles are positioned such that we can assume each vehicle to have approximately the same number of neighbor vehicle d. For such a case we present the result as follows.

Result: Let the protocol given by algorithm 1 be running on N vehicles forming a connected network such that each vehicle has the same number of neighbor vehicles d with which it can communicate. The expected number of cluster heads H_{t_f} formed by the protocol at a threshold t_f where $0 \leq t_f \leq 1$, is given by

$$H_{t_f} = N\left[1 - \sum_{v=0}^{\lceil t\rceil - 1} \binom{d+1}{v} p^v (1-p)^{d+1-v}\right] \tag{1}$$

where $p = 1/(d+1)$ and $t = t_f \times (d+1)$

Proof. The protocol in the algorithm 1 makes a vehicle cluster head when the vehicle gets votes from its neighbor greater than or equal to a threshold t. A vehicle **B** votes for a vehicle **A** if vehicle **A** has the maximum contacts (neighbors) in **B**. If there are two or more vehicles in **B** neighbor list with the same number of maximum contacts then a vehicle is chosen randomly for casting the vote. A vehicle can also vote for itself. If d is the number of neighbor vehicles of a vehicle then a vehicle can get maximum of $(d+1)$ votes. The total possibilities of votes are thus $(d+2)$ i.e., 0 vote, 1 vote, ..., d votes, $(d+1)$ votes.

Let p be the probability that a specific vehicle **A** is voted by one of it neighbor vehicle **B**. Since **B** has $d+1$ options to cast its vote out of which voting to **A** is one of them, we thus have $p = 1/(d+1)$. As we have assumed that all vehicles have the same number of neighbor vehicles d, p is same for all vehicles. It is thus easy to see that the probability that a vehicle i is voted exactly v times by any one of its d neighbors and also by itself, is given by

$$P[\text{vote}_i = v] = \binom{d+1}{v} p^v (1-p)^{d+1-v} \tag{2}$$

The probability that the vehicle i becomes a cluster head i_H is the probability that the vehicle is voted at least t times i.e.,

$$\begin{aligned} P[i = i_H] &= P[\text{vote}_i \geq t] \\ &= 1 - P[\text{vote}_i \leq (t-1)] \end{aligned}$$

The second term on the right hand side is cumulative probability of $P[\text{vote}_i = v]$. We can thus write

$$P[i = i_H] = 1 - \sum_{v=0}^{t-1} P[\text{vote}_i = v] \tag{3}$$

Let the threshold t be written in terms of threshold fraction t_f given by $t = t_f \times (d+1)$ where $0 \leq t_f \leq 1$, as in the line 44 of algorithm. In such a case t will be a real number. The probability $P[i = i_H] = 1$ for $t_f = 0$ since due to 0 threshold t each vehicle i will be a cluster head. The $P[i = i_H]$ for $t_f > 0$ remains same for t intervals $]0,1],]1,2], ...$ which is same as taking ceiling of t i.e., $\lceil t \rceil$.

From above and equations 2 and 3 the probability that any vehicle i is a cluster head can thus be written as

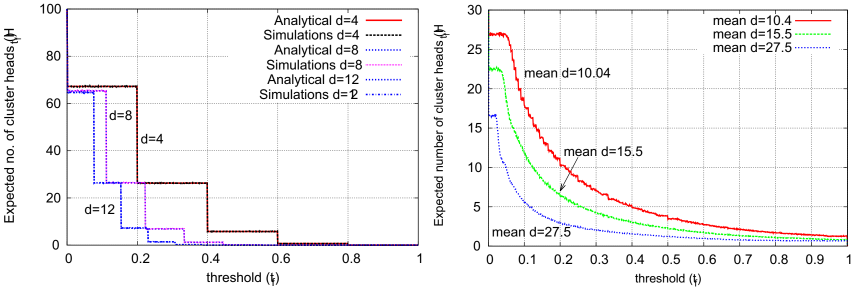
$$P[i = i_H] = 1 - \sum_{v=0}^{\lceil t \rceil - 1} \binom{d+1}{v} p^v (1-p)^{d+1-v} \tag{4}$$

Since the probability of being a cluster head H is same for all vehicles since we have assumed that d is same for vehicles, the expected number of cluster heads H_{t_f} at a threshold t_f is given by

$$H_{t_f} = N \times P[i = i_H]$$

From equation 4, the above equation leads to equation 1.

To validate the analytical result we developed a simulator for the cluster head formation part of the protocol. To have a topology in which all vehicles have the same number of neighbors, we placed the vehicle in a grid in such way that edge vehicles are neighbors of opposite edge vehicles thus forming a closed grid. Such a topology is far from real but nevertheless, since all vehicles have the same numbers of neighbors, it helped to do the analysis. Moreover generalization of the analysis to a case where d is variable can be done following the same approach. Each value of H_{t_f} is calculated by taking an average of 1000 iterations. Fig. 1(a) shows six plots, analytical and simulation for $d = 4$, $d = 8$ and $d = 12$ for 100 vehicles. We see that the analytical and simulated plots completely overlap, thus validating the analysis. We note that the plots are step function. This is because the probability given by equation 2 is a step function depending on d.



(a) Vehicles with constant no. of neighbors validating analysis (b) Vehicle at uniform random locations

Fig. 1. Expected number of cluster heads as a function of threshold for 100 vehicles

The number of steps in the plots are $(d+2)$, as discussed in the proof. These steps are quite notable from the data but not much visible from the plots since the steps are very close to each near higher values of t_f. We also note that for higher d the plot is shifted left. This can be understood from equation $t = t_f \times (d+1)$. As d increases, t_f should decrease to have same t and H_{t_f}.

The importance of the analysis is that it gives us an insight about the cluster head selection algorithm and how the number of cluster heads vary with the threshold. This understanding can help us to understand the protocol behavior in more realistic topologies, which are otherwise difficult to model. Fig 1(b) shows simulation plots of the expected number of cluster heads formed in a real scenario by randomly placing 100 vehicles on a 1000x1000 area. Each point was plotted after 1000 iterations. We had 3 plots for $0 < t_f \leq 1$, with a mean d by changing the transmission radius. Note that for $t_f = 0$ the number of cluster heads will be N. For threshold $t_f > 0$ the step size in the plots is very small. This can be explained by considering the analysis done before. As we know that the number of steps and hence step size in the plots depend on d, in randomly placed vehicles the neighbor distribution d has a wide range of values thus increasing wider range of possibilities of H_{t_f}. Similarly the shifting of plots to left for higher values of mean d can be explained due to the reason described earlier for the plots in Figure 1(a). The result obtained for the random placement of vehicles can be used to tune the protocol in a real scenario to get the best compromise between number of cluster heads and accuracy of display of traffic congestion by setting the parameter t_f.

5 Simulation Setup

We have simulated the proposed protocol using network simulator NS2 version 2.35 [1]. We have used IEEE 802.11p protocol for V2V communication. The transmission range of each vehicle is 250m. For each packet, we set $ttl = 30$ hops. We have done simulations for a straight road of 5 km and with different

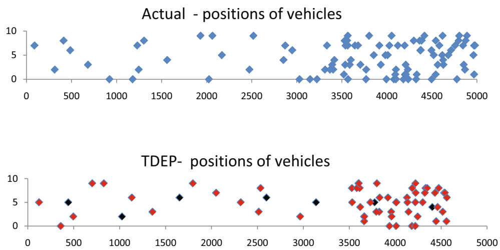


Fig. 2. Display of actual 100 vehicle positions and positions computed from TDEP as seen by a vehicle at $x = 0$

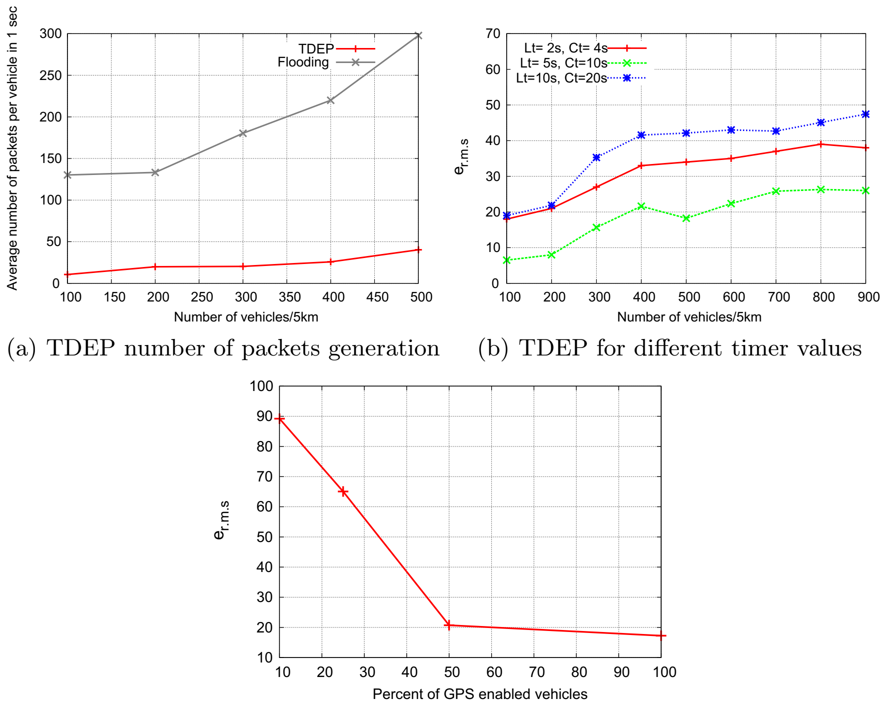
traffic densities scenarios. The vehicles have different velocities ranging from 15 km/hr to 60 km/hr and can overtake each other, thus may form clusters. We allow simulations to take place for 2 minutes and then measure the actual positions of the vehicles in the network. To analyze the accuracy of the protocols we divide the road into segments and measure the root mean square error of all segments. Formally if dA_j is density of jth slot in the actual network and dE_j is the estimated density of jth slot, we define mean square error e_{rms} of difference of densities of all the slots as follows

$$e_{rms} = \sqrt{\sum_{\forall j}(dA_j - dE_j)^2} \tag{5}$$

6 TDEP Characteristics

In the following we describes some of the important characteristics of the protocol obtained from the simulations.

Visualization of the Vehicle Density. The proposed protocol aims to display a map of traffic density to the driver. The purpose is to give driver a close approximation of the actual situation and not to display the exact position of vehicles since the drivers are not interested in the actual position of the vehicles. They are only interested in knowing where there is likely to be more traffic in real time. Each vehicle is shown by a point. More traffic is represented by more points per unit area on the map. When drawing map, the protocol takes the accurate positions of the cluster heads. The position of all other vehicles is approximated with respect to their nearest cluster heads. Figure 2 shows the positions of vehicles as calculated by TDEP. These positions are seen from a vehicle at zero x-axis. We see that the results produced by TDEP are good enough to give a good estimate of the traffic density. The actual error in the representation is calculated by equation 5. This error in display under different conditions is discussed as follows.



(a) TDEP number of packets generation

(b) TDEP for different timer values

(c) TDEP for different percent of GPS enabled vehicles

Fig. 3. TDEP performance characteristics

Number of Packets Generated. Figure 3(a) shows the number of packet generated per second per vehicle for different vehicular densities. The plots show a comparison between a flood based protocol, in which each node forwards packets and TDEP. We see that TDEP generates far less number of packets. This is clearly due to the aggregation property of the cluster based protocol. The interesting fact is that the number of packets generated by TDEP are not much effected by higher vehicular densities. This can be explained with the help of the analysis done in section 4. In the plots of Figure 1(b) we see that for $t_f = 0.5$, when $d = 10.4$, $H_{t_f} = 3.4$, when $d = 15.5$, $H_{t_f} = 2.2$, and when $d = 27.5$, $H_{t_f} = 1.2$ that is, the increase in the vehicular density d is compensated by a decrease in the expected number of cluster heads consequently resulting a less increase in the rate at which packets are generated.

Effect of the Varying Protocol Timers. Figure 3(b) shows the error for TDEP protocol for different values of the timer and densities. We note that the error is higher in case of high or low values of the timers. This is due to the fact that for small time intervals the information is updated more frequently thus generating large number of packets which result in collisions and drop of packets and thus loss of information. On the other hand, when timer values are large the current positions of the vehicles are not updated timely thus introducing error and displaying a density map to the user which is far from real. We see that the best is obtained when $Lt = 5$ and $Ct = 10$.

Effect of Decrease in GPS Information. Figure 3(c) shows the plot when GPS information decreases from 100% to 10%. For the simulations we selected the timer values that gave comparatively better results as discussed previously, that is, $Lt = 5s$ and $Ct = 10s$. The plots of Figure 3(c) shows that the accuracy of the protocol is reduced when the GPS information becomes less available. The interesting point is that the accuracy is not much effected until the GPS enabled vehicles reduce to 50%. This can be explained by considering the fact that cluster head not only aggregate number of neighbor vehicles but also their GPS information. When cluster head or any one its neighbors have GPS information, then during aggregation this can compensate for all those neighbors who do not have GPS. Thus even if 50% of vehicles do not have GPS, their positions can still be approximated with the help of neighboring vehicles and error in displaying the position of cluster head is not much effected.

7 Conclusion and Future Work

In this paper, we have proposed a Traffic Density Estimation Protocol using Vehicular Networks which first selects some vehicles as the cluster heads by a voting mechanism. These cluster heads aggregate the information about the vehicles in their transmission range and then select few vehicles for forwarding this information to the rest of the network. We have simulated our proposed protocol using NS2 network simulator. Simulations shown that the proposed protocol give fairly accurate results under different road traffic scenarios. The proposed protocol is better able to take advantage of GPS enable vehicles. As the number of GPS enabled vehicles increases, the accuracy of the the proposed protocol is also increased. We have also done mathematical analysis of the cluster head selection part of the protocol and have analytically determined the estimated number of cluster head formed.

In future we plan to simulate the protocol on curved roads also having multiple road crossings and in real map scenarios. We also plan to see the effect of changing the transmission range on the accuracy of the protocol.

Acknowledgements. This work was partially funded by grant 'Progetti di Ricerca, La Sapienza', prot. C26A129S3Y, 2012.

References

1. Network simulator ns-2, `http://www.isi.edu/nsnam/ns/`
2. Ababneh, N., Labiod, H.: Safety message dissemination in vanets: Flooding or trajectory-based? In: Med-Hoc-Net, pp. 1–8. IEEE (2010)

3. Almahorg, K., Basir, O.: Simulation-based performance comparison of vanets backbone formation algorithms. In: 12th IEEE/ACM International Symposium on Distributed Simulation and Real-Time Applications, DS-RT 2008, pp. 236–242 (October 2008)
4. Hofmann-Wellenho, H.L.B., Collins, J.: Global positioning system: Theory and practice, vol. 4. Springer (1997)
5. Bilstrup, K., Uhlemann, E., Ström, E.G., Bilstrup, U.: On the ability of the 802.11p mac method and stdma to support real-time vehicle-to-vehicle communication. EURASIP J. Wirel. Commun. Netw., 5:1–5:13 (January 2009)
6. Chang, W.-R., Lin, H.-T., Chen, B.-X.: Trafficgather: An efficient and scalable data collection protocol for vehicular ad hoc networks. In: 5th IEEE Consumer Communications and Networking Conference, CCNC 2008., pp. 365–369 (January 2008)
7. Hsiao, H.-C., Studer, A., Chen, C., Perrig, A., Bai, F., Bellur, B., Iyer, A.: Flooding-resilient broadcast authentication for vanets. In: Proceedings of the 17th Annual International Conference on Mobile Computing and Networking, MobiCom 2011, pp. 193–204. ACM (2011)
8. Huang, D., Shere, S., Ahn, S.: Dynamic highway congestion detection and prediction based on shock waves. In: Proceedings of the Seventh ACM International Workshop on VehiculAr Inter-NETworking, Systems, and Applications, VANET 2010, pp. 11–20. ACM (2010)
9. Ibrahim, K., Weigle, M.: Cascade: Cluster-based accurate syntactic compression of aggregated data in vanets. In: 2008 IEEE GLOBECOM Workshops, November 30–December 4, vol. 30, pp. 1–10 (2008)
10. Lee, E.-K., Yang, S., Oh, S., Gerla, M.: Rf-gps: Rfid assisted localization in vanets. In: IEEE 6th International Conference on Mobile Adhoc and Sensor Systems, MASS 2009, Macau, pp. 621–626 (2009)
11. Nadeem, T., Dashtinezhad, S., Liao, C., Iftode, L.: Trafficview: traffic data dissemination using car-to-car communication. SIGMOBILE Mob. Comput. Commun. Rev. 8(3), 6–19 (2004)
12. Padron, F.M.: Traffic congestion detection using vanet. Master thesis, Florida Atlantic University, USA (2009)
13. Parker, R., Valaee, S.: Vehicle localization in vehicular networks. In: 2006 IEEE 64th Vehicular Technology Conference, VTC-2006 Fall, pp. 1–5 (September 2006)
14. Porikli, F.: Traffic congestion estimation using hmm model without vehicle tracking. In: IEEE Intelligent Vehicles Symposium, pp. 188–193 (2004)
15. Ros, F., Ruiz, P., Stojmenovic, I.: Acknowledgment-based broadcast protocol for reliable and efficient data dissemination in vehicular ad hoc networks. IEEE Transactions on Mobile Computing 11(1), 33–46 (2012)
16. Singh, R.P., Gupta, A.: Traffic congestion estimation in vanets and its application to information dissemination. In: Aguilera, M.K., Yu, H., Vaidya, N.H., Srinivasan, V., Choudhury, R.R. (eds.) ICDCN 2011. LNCS, vol. 6522, pp. 376–381. Springer, Heidelberg (2011)
17. Tonguz, O.K., Wisitpongphan, N., Parikh, J.S., Bai, F., Mudalige, P., Sadekar, V.K.: On the Broadcast Storm Problem in Ad hoc Wireless Networks. In: 3rd International Conference on Broadband Communications, Networks and Systems, BROADNETS 2006, pp. 1–11 (2006)
18. Pattara-atikom, R.P.W., Luckana, R.: Estimating road traffic congestion using cell dwell time with simple threshold and fuzzy logic techniques. In: IEEE Intelligent Transportation Systems Conference, pp. 956–961 (2007)