International Journal on New Computer Architectures and Their Applications (IJNCAA) 1(1): 245-256
The Society of Digital Information and Wireless Communications, 2011 (ISSN 2220-9085)

Cooperative Mechanisms to Carry and Forward Packets in a DTN VANET

Gianni Fenu and Marco Nitti
Department of Computer Science, University of Cagliari, Via Ospedale 72, 09124 Cagliari
{fenu, marconitti}@unica.it

ABSTRACT

The aim of this paper is to find the best strategies to carry and forward packets within VANETs that follow a Delay Tolerant Network. In this environment nodes are affected by intermittent connectivity and topology constantly changes. When no route is available and the link failure percentage is high, the data must be physically transported by vehicles to destinations. Results show how, using vehicles cooperation and several carry and forward mechanisms with different deliver priorities, is possible to improve performance for free in terms of data transmissions. In our experiments we use realistic traffic data obtained by multi-agent traffic simulator and also study which are the best conditions to apply our mechanism in an everyday scenario.

KEYWORDS

VANET; Delay Tolerant Network; Carry and Forward mechanism; Idle Periods; mobility modeling.

1 INTRODUCTION

Vehicular Ad-hoc Networks or VANETs are particular type of mobile networks where nodes are vehicles and no fixed infrastructure is needed to manage connection and routing among them. Vehicles, in a pure VANET, are self-organized and self-configured thanks to "ad hoc" routing protocols that manage message exchanges. These characteristics make this technology a good solution to create applications for safety purposes or simply to avoid traffic congestion. Vehicle's inside devices are also designed to access internet when a gateway is encountered. Road Side Unit (RSU) or Access Point (AP) could be used as gateways in a hybrid VANET to work as intermediaries between vehicles and other networks. Often cars move at high speeds and this behavior reduces transmission capacity, creating issues like:

1. Rapid changes of network topology. The state of connectivity between nodes is constantly evolving.
2. Frequent disconnections. When traffic density is low, distance between vehicles can reach several kilometers beyond the range of the wireless link and this involve link failure that can last several minutes.
3. High nodes congestion in heavy traffic conditions can affect the protocol performance.
4. High level of packet losses. Measurements of UDP and TCP transmissions of vehicles in a highway passing in front of a RSU moving at different speeds, report losses on the order of 50-60% depending on the nominal sending rate and vehicle speed.

5. Identify unambiguously each mobile node using the proper addressing method.
6. Environment obstacles like tunnels, traffic jams, lakes etc. could interfere with the transmission signal.
7. Interoperability with other networks has to be achieved. Nodes must be able to exchange data with other types of networks, especially those based on fixed IP addresses.

For these reasons all standard routing protocols result inadequate to ensure good connection and achieve high performance. In order to obtain suitable routing protocols we have to exploit the characteristics of VANET. An interesting property of vehicles is that they move along roads unchanged for years and this allow recognizing specific mobility patterns (roads). So, knowing position speed and direction of vehicles we can predict their future geographic locations and plan some strategies to deliver the packets exploiting nodes cooperation. This paper is based on a scenario already used in [1] by Fiore and Barcelos, with the difference that we measure how traffic data varies using different deliver priorities. We have introduced in our code a parameter called alpha in order to manage cooperators behavior during deliveries. Alpha, in fact, can affects the choices of possible targets during cooperation and also influences the amount of data delivered during each step. In our scenario nodes can download data from fixed infrastructure scattered in the topology or from other vehicles during cooperation. Infrastructures can be placed in highway or in urban environment and are all connected via backbone to a common server which periodically collects vehicle status information. In particular the server focuses on groups of cars travelling in neighbor RSUs in order to predict with good precision possible meetings. In this way we are able to implement a carry and forward mechanism that exploits time used by vehicles to cross dark areas between different RSUs coverage to increase for free the global throughput.

The main contribution of this paper is:

1. Definition of a Vehicular Ad Hoc Network scenario that opportunistically allows downloading packets when vehicles cross RSU.
2. Definition of RSU idle period and research of how traffic density influence its duration. Results are obtained with simulations executed on data taken from the multi-agent traffic simulator developed at ETH Zurich where the traffic approximates a M/GI/ ∞ queue system (Poisson distribution).
3. Proposal of several scheduling mechanisms that exploit RSU idle periods to organize distribution of packets to specific vehicles called cooperators whose task is to physically carry data toward the final destination. Giving different deliver priorities we discover how to help vehicles to finish their translation faster in order to allow them to help in future cooperation.

The paper is organized as follows: section 2 discuss related work; section 3 describes the vehicular scenario showing the amount of RSU idle periods obtained from simulations; section 4 proposes scheduling algorithms (and related results) in order to benefit from the carry and forward concept; Section 5 offers some conclusions.

2 RELATED WORK

In these years have been proposed several protocols to route data within VANETs and we can group them in two main categories:

1. Topology-based routing divided in:
   - *Proactive* in which topology is constantly updated due to periodical collection of traffic conditions. In this way you can quickly find the destination but many resources are being used during updates.
   - *Reactive* where the path between source and destination is calculated only when requested (usually before a transmission). This method can save time and band when too many vehicles participate to the network.
   - *Hybrid* that is a combination of the previous (proactive for near destinations and reactive for far destinations).
2. Position-based routing or geographic routing in which we need:
   - *A location service* to find the destination. This unit is important because often IP addresses aren't enough to identify unambiguously all the nodes. In these cases it's preferable to use a hierarchic approach in which vehicles are recognized by geographic position and unique ID.
   - *Forwarding Strategies* to send the packet to the destination reliably and as quick as possible. The path between source and destination is constructed step by step by each vehicle.

Actually all the protocols studied can be situated in one of this two categories. The correct protocol must be chosen, considering the features of the network in which we're going to communicate. In our scenario, where there isn't total coverage and transmissions are affected by long delays, the best choice is the geographic routing protocol category combined with cooperation. In particular we focus on opportunistic forwarding strategies, in which nodes schedule the forwarding of packets according to opportunities; [2], [3] e [4]. The opportunity may be based on: historical path likelihoods, [2], packet replication, [3], or on the expected packet forwarding delay, [4]. These scheduling mechanisms are based on epidemic [5] and probabilistic routing [6] and their objective is to optimize contact opportunities between vehicle and RSUs to forward packets in intermittent scenarios. However, these protocols don't consider how to exploit the casual vehicle-vehicle contacts. If we know meetings in advance, we can involve some unaware passerby during communication and let them physically carry data to destination. SPAWN, [7], is a good example of cooperative strategy for content delivery. It works using peer-peer swarming protocol (like bit torrent) including a gossip mechanism that leverages the inherent broadcast nature of the wireless medium, and a piece-selection strategy that uses proximity to exchange pieces quicker. We assume that our scenario use similar SPAWN based mechanism to improve the distribution of popular files among vehicles. Imagine, for example, a VANET where a group of nodes try to download the first page of the local newspaper sharing chunks of information when they meet. Unique difference between two scenarios is that SPAWN considers unidirectional traffic over highways while we consider a more complex urban environment.

3 RSU IDLE PERIODS AND SUITABLE CONDITIONS

In this section we describe the simulation scenario introducing the concept of RSU idle periods calculated with different traffic densities. Analyzing the results obtained by placing the RSUs in different spots we identify the best conditions to perform our Carry and Forward (C&F) mechanism.

Like said before in our scenario vehicles can download information from fixed infrastructure or from RSUs located along roads. RSUs are connected via backbone and scattered among the topology but they don't cover the whole paths followed by vehicles. When a vehicle reaches the RSU coverage for the first time obtains identification (Node-ID) and then starts to periodically broadcast its direction, speed and ID. These beacons of information are forwarded to the server that gets a constantly updated overview of traffic under RSUs. So we only know status of vehicles under coverage but it's possible to obtain these information also when they travel in dark areas using historical paths.

TCP / IP stack protocols don't provide an high data transfer to vehicles due to the harsh physical conditions in which they have to communicate, however RSUs can use them to send data to the server exploiting the wired and more reliable connections. To manage the link failure, the packets losses and the large delays during communications the RSUs are provided of storing and computing capabilities as happens in Delay Tolerant Network (DTN), [8]. If some packets are lost, RSUs don't perform retransmission immediately but wait until the entire block of data has been transmitted. All missed packets are retransmitted together optimizing in this way the use of band. The server uses vehicles status (speed, direction, id) to choose how to manage data distribution among RSU. When a RSU receives data and meeting predictions from server, starts to deliver packets to consumer under coverage and to organize cooperation for vehicles out of coverage. Cooperation is achieved scheduling part of RSUs data among cooperators that have high probability to encounter receivers during their trip. In highway scenarios, in which vehicles follow the same direction for long periods, the server predicts without doubts which will be the next RSU on the path. From now on we will use specific terminology to refer properly to actors in the network:

- *Consumer* is a vehicle that downloads whenever has the opportunity (from RSUs or other vehicles).
- *Receiver* is a consumer that is designed to receive data from cooperators. It is discovered by the C&F mechanism. Consumer usually becomes receiver only if has high probability to meet cooperators during its trip.
- *Cooperator* is a common vehicle that isn't interest in download files but can be used by RSUs to carry packets for receivers.

- *Idle period* is the time's slot in which the RSU has no consumers under coverage. RSU isn't really idle because it's busy to manage cooperation between cooperators but for simplicity we continue to use this term.
- *Dark area* represent the stretch of road between two coverage.

Normally consumers can only receive data when are under a RSUs coverage or when encounter other consumers with chunks of information that they miss. However when they leave coverage, they have to wait to reach the next RSU before to resume their download (Fig. 1).

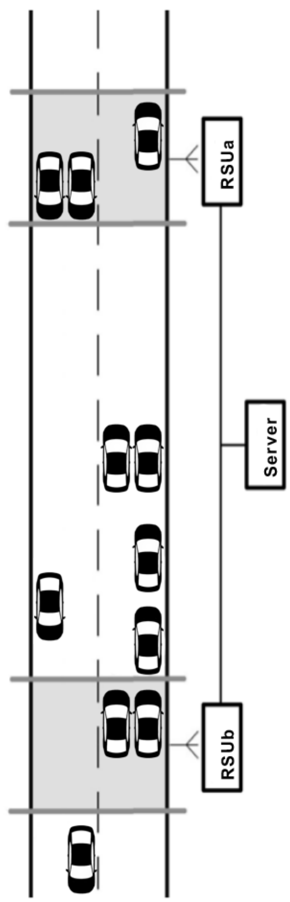

Fig. 1. Network scenario

Using our mechanism instead, with a correct study of topology and an optimized packets distribution, we can exploit the dead time spent by vehicles crossing dark areas to deliver chunks of information trough opportunistic cooperation with cars not directly involved in communication. Cooperators will be able to meet consumers during their trip in dark areas and deliver to them the data that they are carrying.

Our simulations work on selected real-world road topology data from the area of Zurich, Switzerland, due to the availability of large-scale microscopic-level traces of the vehicular mobility, [9]. We didn't use traditional network simulator, such as ns-2, due to the elevate number of vehicles reported in our trace. Instead, we prefer to use Matlab that has optimized functionality to manage operations between matrixes. In each experiment, before calculate idle periods, we have to set three parameters: (i) RSUs position (the choice can be made based on traffic density or environment conditions), (ii) δ or consumers density and (iii) the range of RSUs coverage. With these three parameters we can create, with the same traces, several frameworks in order to see the behavior of RSUs under different conditions. In particular the most important one is δ because allow us to set the percentage of vehicles that try to download some content. Since the number of vehicles is fixed for each simulation this can greatly influence the final results.

For each second of simulation, we checked how many consumers travel under coverage and if RSU is busy to deliver data to them. Finally we repeat the experiment several times incrementing the coverage range up to a maximum of 300 mt in order to involve every time a bigger piece of road. Fig. 2 shows the amount of idle time (the sum of all periods) calculated with different δ in a RSU with 300 mt coverage; in x-axis we can see the consumers density while in y-axis the percentage of idle period (scaled from 0 to 1).

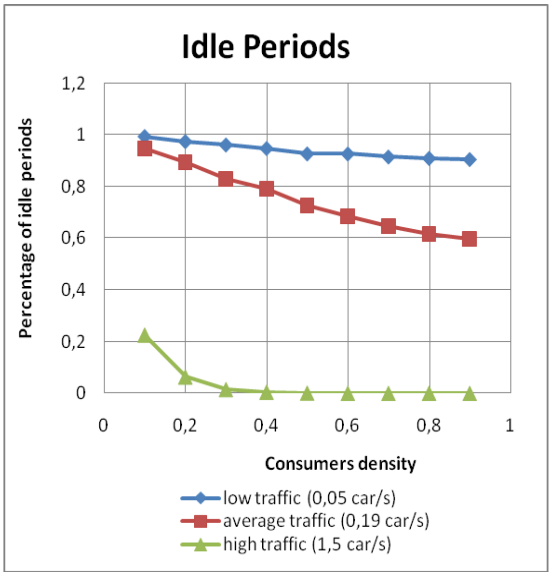

Fig. 2. Idle periods calculated with different traffic density.

As we can see, with low traffic density (0.05 car/s), the RSU is almost always idle and remains free for about 88% of the simulation. In areas with average traffic density (0.19 car/s) results show a considerable amount of time usable by scheduler to manage cooperation among vehicles. A steady stream of cars instead (1.5 car/s), involves intense activity of the RSU, which, even with low consumer density, remains busy to transmit data to consumers. Note that, in this last case, the time available for the scheduler becomes zero quickly and apply C&F algorithm becomes impossible. However, the idle period alone isn't enough to achieve cooperation . In fact, if there isn't at least a cooperator under the idle RSU or if there isn't at least a receiver available to receive data the mechanism can't be used and the time is wasted. It's also necessary that cooperator and receiver travel in opposite directions in order to predict the meeting.

So we can distinguish between generic idle periods and usable idle periods. The scheduler works only during usable idle periods, but sometimes the necessary conditions to obtain it, rarely occur. The RSU transmission range, the average vehicles speed, the consumer density and the distance between RSUs all influence the occurrence of usable idle periods and for this reason we have to place the infrastructure very carefully. Performing experiments we notice that, in zones with a medium/high traffic flow, consumers density value between 0.3 and 0.5 and average speeds (about 20-30 km/h) the chances to apply the mechanism are relatively high.

4 SCHEDULE STRATEGIES AND RESULTS

In this section have been tested several mechanisms for scheduling packets in opportunistic C&F protocol in order to optimize network performance. Mainly three techniques have been proposed:

1. Distribute the available data carried by cooperators equally among consumers.
2. Give greater priority to vehicles which have almost finished downloading their file.
3. Designate as receivers of the packets only vehicles that have more probabilities to meet cooperators.

The algorithm that implements these techniques examine the traffic second after second. Every moment consumers and cooperators state is updated by using two data structures and all RSU are checked to find out which ones are free and which ones are busy. Only consumers that travel in dark areas are labeled as receivers during a particular second. For each of them, data structures are update with following information: RSU target that is the RSU where the consumer is directed to (or where we estimate it is directed), x and y coordinates, direction and finally the file status that represents how many bits have been downloaded so far by the vehicle. Similarly cooperators have a data structure that is updated during each second of usable idle period with this information: RSU source that is the RSU where cooperator is coming from, x and y coordinate, direction, a list of possible receivers, another complementary list to the previous one with the amount of data destined to each receiver (transaction list) and finally a TTL (time to live) counter used to measure the lifetime of data carried. Once updated the two structures, each cooperator is able to check its receivers list to see if someone is close enough to its position to establish a connection. If this occurs data are transferred in the amount indicated by the transaction list. The data structures are partially adjusted after each meeting and totally updated when vehicles encounter a new RSU.

Like said in the previous section this mechanism works at an high abstraction level, above data-link layer of the TCP/IP stack because we are only interested in understand if the global scenario performance could be improved. For this reason we assume that all transmissions occur instantly without any problems related to packet losses or environmental interferences in signal. Regarding physical and data-link protocols we can suppose to use the well-know 802.11p standard. Amount of data transferred during each encounter is fixed and is based on the average link durations (around four seconds). If a vehicle finishes to download its files, it is automatically deleted from the list of consumers, and can be candidate to become a cooperator. A cooperator can carry only a predetermined amount of data, so it's better to decide in advance how to divide this amount among receivers. The division strategy, managed by scheduler, depends on the value of a parameter that we call α: if α is equal to 0 means that all data carried must be delivered only to the receiver which have the most advanced file status (maximum priority) while if it is equal to 1 means that data must be divided equally among all receivers (equal priority). Thus, by simply changing the α value we can determine the percentage of consumers to which give higher priority (if $\alpha = 0.2$ means that only 20% of the receivers with bigger file status will receive data). For the third strategy instead we have to calculate the probability that two vehicles meet themselves during their trip, so is necessary to know, for each pair of RSU, the percentage of vehicles traveling form the first to the second one and vice versa. To do so we use the historical paths information available from simulator data combined with the study of traffic streams (Fig. 3). This isn’t a novel method to calculate meetings but it is for sure more simple than other approaches. In future studies, other solutions can be proposed. For example, could be very interesting to use navigator GPS information to know vehicles destination and hypothesize which roads will be drive, using Dijkstra algorithm or studying traffic congestion. Another method consists in perform a census to know generic drivers’ behaviors for each day and hour of the week in order to calculate vehicles streams.

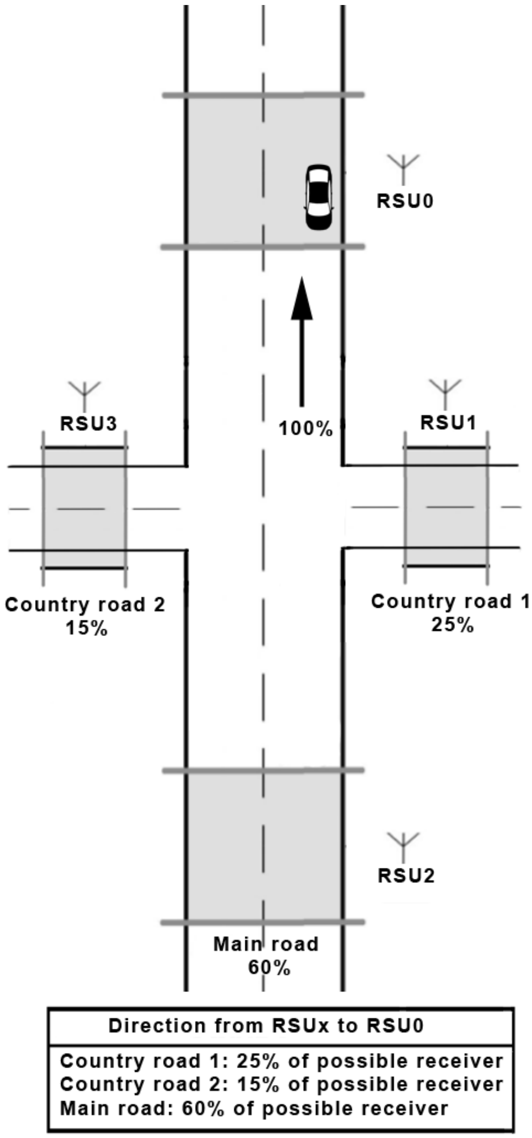

Fig. 3. in this example we assume that RSU0 use traffic stream percentage to decide how to schedule data. 60% of available packets were prepared for receivers coming from RSU2, 25% for receivers from RSU1 and other 15% for receivers from RSU3. Then data were divided properly among cooperators.

At this point, we only have to decide if our target is to optimize data transfer or ensure equity during packets distribution. If we try to optimize performance, the scheduler have to divide packets only among cooperators headed to road with high vehicles stream. However, in this way, all consumers that travel in low traffic zones will remain isolated. To avoid this situation, we use traffic flows to divide in percentage the data scheduled for each roads, in order to give a connection chance to all consumers. The most important results were obtained from two large experiments:

1. A simulation using four RSUs placed in particular spots. Between each pair of RSU there aren't crossroads or bifurcations but only straight road and this situation give us the security to know in advance all possible meetings between vehicles. This experiment aims to verify the proper functioning of the mechanisms proposed.
2. A simulation using three RSUs placed casually on the map. This is a more realistic scenario that allows us to see if protocol works properly in harsh conditions too.

Table 1. Simulation input parameters.

	Experiment	
	First	Second
Nr. RSU	4	3
RSU bit/s	10 Mb/s	10 Mb/s
File size	40 MB	10 MB
Car range	200 mt.	200 mt.
TTL	60	300

Table 1 shows simulations input parameters. In particular, “file size” describes the amount of data that each consumer has to download. Fig. 4 and 5 instead show experiments results given in terms of MB delivered respectively from RSUs and from cooperators.

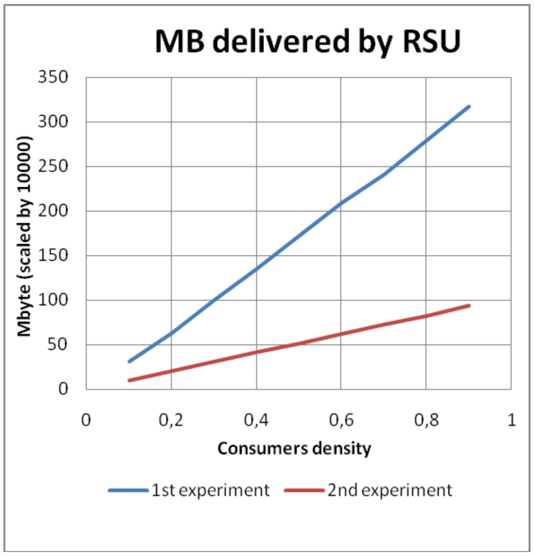

Fig. 4: Data delivered by RSU.

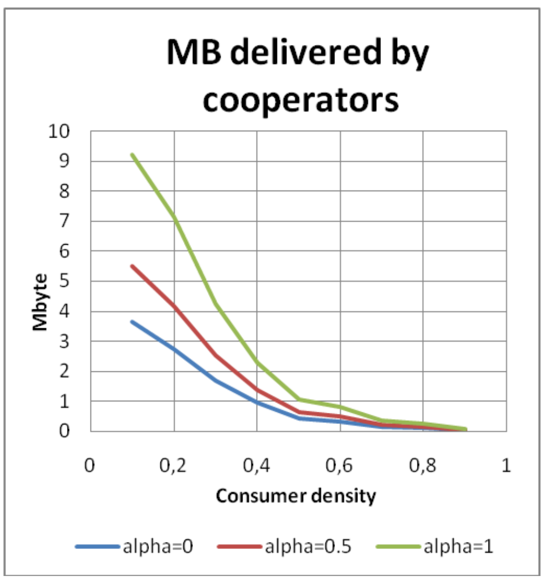

Fig. 5: Data delivered by cooperators.

Analyzing results, you may notice that high percentage of packets is handed over by RSUs and only a small amount due C&F protocol. However, this small amount helps vehicles to finish their downloads faster improving, indirectly, network performance and effectiveness of cooperation. Since RSU manages most of the packets, it‘s obvious that the amount of data distributed globally in the system increases with increasing the consumers density. In first experiment with α=0 the system delivers from a minimum of 306 GB to a maximum of 3 TB and 177 GB (in three hours of simulation from 4 RSUs). Instead, packets distributed by cooperators decreases with increasing value of δ. This behavior was predictable because:

- The scheduler is busier when consumers increase and consequently less usable idle periods are available to organize cooperation.
- More consumers mean fewer cooperators because vehicle's number is fixed.

Moreover, in fig. 5 we can note how increasing α for smaller values of δ, the number of packets delivered increases. This means that using a low priority during distribution of data among receivers produces, in terms of performance, more acceptable results. However, if our intent is to increase number of files completely downloaded then it's preferable to set a lower value of α (so we maximize priority). First simulation in fact shows that the amount of files completely downloaded rise proportionally to priority.

Second experiment instead return some different results. Without knowing in advance the route taken by vehicles, we must assume, through a probabilistic calculation, which will be the target RSU for each consumer. Based on these assumptions (which could be wrong) we calculate for each cooperator the receivers list. For this reason in this second experiment, performances are worst respect the previous one but the protocol behavior is quite similar. The unique difference is that, in this case, the number of file completed doesn't raise proportional to priority. This happens because the algorithm only attempts to predict possible encounters that sometimes may not occur. All missed meetings result in lost opportunities to increase the overall efficiency of the network. Moreover, since RSUs are farther respect to the first experiment, this forced us to set a TTL value high enough to ensure that all vehicles have opportunity to meet. So, when the meeting doesn't happen, the cooperator may remain several seconds wandering on the map, before being used again for other receivers (provided along the trip encounters another free RSU). This situation is more accentuated when the topology has a too homogeneous traffic distribution because we are unable to detect the busier roads and, even using historic path, predict chances of meeting become hard.

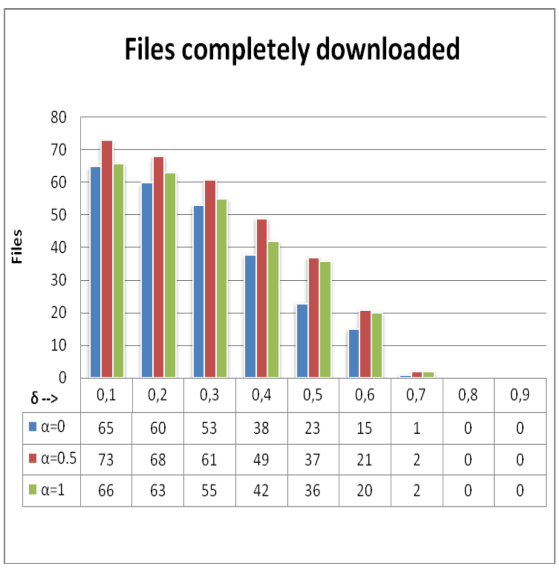

Fig. 6: Histogram of files completely downloaded. Best choice, for low δ values, is α=0.5.

For this reason is better to place the RSUs always at principal city crossroads, especially in main streets or in highway (where only two directions exist). Finally fig. 6 shows the amount of file completed using different priority value.

For the same reason given above, random scenario produces interesting behaviors calculating files downloaded, especially for lower values of δ. In fact, as we can see, using average value of α (0.5) instead of high value (0) we can complete more files hoping, in this way, to increase future cooperation. A medium priority allow us to define for each cooperator a bigger list of receivers respect to the list obtained with higher priority (where we privilege only the percentage of vehicles with more advance file status). In this way cooperators have more chances to encounter at least one receiver of their lists and increase the overall amount of data delivered. It's important to remark that this approach enhances cooperative content sharing in VANETs without introducing additional overhead since we only use RSUs idle period to manage the scheduling process. Our intent is to improve this mechanism and conduct further experiments, increasing simulation duration and RSUs number in order to find out if cooperation's level increases in longer simulation, influencing positively VANET performance. Moreover using a unit of time equal to one hundredth of second is also possible to obtain larger idle periods. We are also interested in adopting a more advanced simulation platform like one described in [10] in order to facilitates the dynamic interaction between vehicles and RSUs.

5 CONCLUSIONS

In this paper has been proposed a vehicular framework that allows vehicles to opportunistic download packets when cross RSUs. The scenario adopts some feature from the Delay Tolerant Network, giving to RSU storing and computing capabilities to manage delays, and benefits of a Carry and Forward mechanism. Using this protocol is possible to increase the global throughput of a real scenario due to the exploitation of RSUs idle periods. If traffic conditions, vehicles speeds, vehicles distribution and consumers density are balanced the increment of performance can be relevant. Then we also explain why big idle periods don't always mean time usable by scheduler. In fact if a RSU is idle but no cooperators are available for receive data to carry or no receiver is detected, this time results wasted. With this assumption we propose different strategies to schedule packets and change the protocol operation, producing different results. If our application requires the urgent delivery of some packets to a particular vehicle, we should use a high priority delivery strategy, while if the goal is to maximize the number of data sent it's better to use an equal priority delivery strategy. These behaviors were tested in two different simulations. We discover that results strongly depends by the position chosen to place the RSUs and by the chances to predict properly the meetings between vehicles. In fact within an ideal scenario, where prediction are precise, it's possible to choose the strategy based on preferences (maximize data transfer or number of files completed) while in a random scenario we must avoid to use high priority. With high priority strategy, in fact, we places too much trust in meetings that may not occur losing the opportunity to deliver at least some chunk of data. With a moderate priority instead (like α=0.5) is possible to obtain more balanced results.

6 REFERENCES

1. Fiore M., Barcelo-Ordinas J.M.: Cooperative download in urban vehicular networks. In: IEEE 6th International Conference on Mobile Adhoc and Sensor Systems, pp 20--29. MASS '09 (2009).
2. Burgess J., Gallagher B., Jensen, D.; Levine B. N.: MaxProp: Routing for Vehicle-based Disruption Tolerant Networks. In: 25th conference on Computer Communications, pp. 1--11. INFOCOM (2006).
3. Balasubramanian A., Levine B.N., Venkataramani A.: DTN Routing as a Resource Allocation Problem. In: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications, Volume 37, Issue 4, pp. 373—384. ACM SIGCOMM '07, New York (2007).
4. Zhao J., Cao G.: VADD: Vehicle-assisted data delivery in vehicular ad hoc networks. In: 25th IEEE International Conference on Computer Communications, pp. 1--12. IEEE INFOCOM, Barcelona, Spain, (2006).
5. Vahdat A., Becker D.: Epidemic routing for partially connected ad hoc networks. Technical report, Duke University (2000).
6. Doria A., Lindgren A., Schelén O.: Probabilistic routing in intermittently connected networks. In: SIGMOBILE Mobile Computing and Communication, Volume 7, Issue 3, pp. 19--20. ACM (2004).
7. Das S., Nandan A., Gerla M., Pau G., Sanadidi M.Y.: Cooperative downloading in vehicular ad-hoc wireless networks. In: Second Annual Conference on Wireless On-demand Network Systems and Services, pp 32--41. WONS (2005).
8. Fall K.: A delay-tolerant network architecture for challenged internets. Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, pp. 27--34. ACM, SIGCOMM '03, New York (2003).
9. Burri A., Cetin N., Nagel K.: A large-scale agent-based traffic microsimulation based on queue model. In: Proceedings of Swiss transport research conference (STRC), pp. 3--4272 Switzerland (2003).
10. Yang Y., Bagrodia R.: Evaluation of VANET-based advanced intelligent transportation systems. In: Proceeding of the sixth ACM international workshop on

VehiculAr InterNETworking, VANET '09, pp. 3--12, Beijing, China (2009).