A Map-Based Sensor Data Delivery Protocol for Vehicular Networks

Sergio Martínez Tornell, Carlos T. Calafate, Juan-Carlos Cano and Pietro Manzoni
Universitat Politècnica de València
Camino de Vera, s/n, 46022 Valencia, Spain
sermarto@upv.es, {calafate, jucano, pmanzoni}@disca.upv.es

***Abstract*—Nowadays our vehicles incorporate multiple sensors that collect information not only about speed or fuel consumption, but also about weather (rain sensors), road state (ESP sensors), etc. Achieving a distributed traffic monitoring system that combines all this information gathered by different vehicles is a challenging task. In particular, we consider that vehicular ad-hoc networks (VANETs) are particularly suitable for this endeavor. VANETs present some characteristics such as a high mobility, a variable node density or frequent radio obstacles that are prone to generate network partitions. Under these conditions, a complete multi-hop path between the sender and the receiver of a message is rarely available. Previous studies have proposed using DTN protocols to address these problems. In this paper we propose a collecting sensor data routing protocol, called Map-based Sensor-data Delivery Protocol (MSDP), which combines information about the road map and the nodes' future routes to improve data delivery. Through simulations based on accurate mobility and propagation models, and using the well-known Epidemic protocol as reference, we demonstrate that our proposal is able to significantly reduce channel usage, while maintaining or even improving the packet delivery ratio.**

***Index Terms*—delay tolerant networks, Wireless sensor networks, wireless networks, VANET, GPS.**

I. Introduction

Wireless networks have evolved at a very fast rate, being applicable to several contexts and different communication solutions. For the automobile industry, many wireless solutions have been proposed to improve safety-related and data communication among vehicles and between vehicles and infrastructure. Concerning safety-related communications, vehicles have different sensors which collect information not only about engine status, or speed, but also context information (e.g. weather or traffic status). The collected information could be sent to data centers inside the backbone network through Vehicular Ad-Hoc Network (VANET) technologies and processed to improve road security and traffic management.

VANETs are considered a subset of Mobile Ad-hoc NETworks (MANETs). However, the high speed of the nodes in a VANET, and the presence of obstacles like buildings, produce a highly variable network topology, as well as more frequent partitions in the network. Typical MANET protocols do not adapt very well to these conditions since a complete connected path between sender and receiver is usually missing. Under these conditions, Delay Tolerant Networks (DTNs)[1] are an alternative able to deal with VANET characteristics.

DTNs allow sharing information between nodes even in the presence of high delays. When the DTN concept was introduced, it was proposed as a solution for the *Interplanetary Internet*, a kind of internet-like network between satellites. DTN follows a scheme known as *store, carry, and forward*. In DTN, when a message cannot be routed to its destination, it is not dropped; instead, it is stored and carried until a new route becomes available. This mechanism can be applied to VANETs to take advantage of the high degrees of mobility [2].

In this article we propose the Map-based Sensor-data Delivery Protocol (MSDP), a DTN routing protocol that combines information obtained from the Geographic Information Service (GIS) with the actual street/road layout obtained from the Navigation System (NS) to find the best route. Thus, the novelty of our proposal is that our protocol is not based simply on geographic positions or distances, but routing decisions consider the programmed route of the vehicle, the expected time to reach the message's destination, the amount of data stored in the vehicle's buffer, the amount of data expected to be exchanged with the destination, as well as the degree of trust of the source for all the considered information to efficiently deliver sensor data messages to the control center. Using accurate mobility and propagation models, we thoroughly evaluate our protocol against the well-known Epidemic protocol. The results show that MSDP achieves a higher delivery probability and a smaller average delay with much lower channel congestion levels.

This paper is organized as follows: in Section II we refer to some of the most relevant proposals in this field. Later, we describe our proposal in Section III. Our simulation environment and settings are presented in Section IV. In Section V we expose the chosen metrics and the obtained results. Section VI concludes this paper and provides details about future work.

II. Related Work

In the last years DTNs have been proposed as a suitable solution to address some of the limitations of classical MANET protocols when applied to VANET environments. Several authors have presented different protocols and solutions. In this article we have divided them into two groups, depending on whether they collect external information or not. In the first group we include the Epidemic Protocol [3]. In this protocol, nodes announce their presence periodically. When a new neighbor is detected both nodes interchange

their *received messages list*. From that moment on, nodes exchange messages with their neighborhood in an attempt to make sure that all neighbors get a copy of the messages they are carrying. The multi-copy scheme leads to a waste of resources when the number of nodes increases, as in the case of VANETs. To reduce the amount of generated network traffic some modifications have been proposed; in [4] and [5] the authors propose limiting the number of copies. In order to keep a high delivery ratio, they try to choose the best node to send a copy to; however, they do not take advantage of multi-hop communication. In [6] the metric used to determine which will be the next forwarding node is a parameter obtained from the list of previous encounters of a node, *i.e.* nodes with more previous encounters are considered better forwarders. There have been some improvements, like PRoPHET+ [7], that also consider some general parameters such as buffer space availability, energy consumption, etc. However, we believe that the use of the previous encounters to determine the forwarding node is not applicable for the application studied in this paper because, although human mobility patterns tend to be repetitive due to their daily routines, using the list of previous encounters as a routing parameter would produce a high delay.

Within the group of protocols that use external information collection, several solutions and schemes have been proposed. Some authors have focused on protocols that make an intensive use of deployed infrastructure [8]. The consequence is a huge deployment cost. Some other authors have focused on the direction of nodes as a decision parameter [9], obviating that in VANETs node mobility is constrained by street topology, and so the direction of the nodes, for urban environments in particular, is rarely going to be constant. In this context GeOpps [10] is one of the most advanced protocols proposed for DTN in VANETs. It uses the information obtained from the NS to determine the closest point to the destination along the route of a node. The next forwarding node is the one whose route passes closer to the destination; this proposal has some problems that have been partially solved in GeoDTN+NAV [11]. GeoDTN+NAV adds a preceding step: before starting the DTN routing, it tries to find a path using GPSR [12]. However, none of these protocols considers important parameters like the amount of data in the forwarding node's buffer or the amount of data that the forwarding node will be able to interchange with the destination, neither do they consider the need for fragmenting messages when their size surpasses the Maximum Transfer Unit (MTU) supported by the network. Moreover, GeoDTN+Nav was evaluated using the Free Space propagation model [13], which guarantees that every node inside the transmission range, in the absence of interferences, will receive the message correctly. This condition, which is far from being real [14], benefits the greedy forwarding scheme used by GeoDTN+Nav, producing too optimistic simulation results.

III. Protocol Description

We have designed a protocol called Map-based Sensor-data Delivery Protocol (MSDP) with the purpose of collecting information from a vehicular sensor network and delivering that information to a control center located inside the backbone of the network. The information is obtained from sensors deployed in vehicles which can be retrieved using, for example, an On Board Diagnostic 2 (OBD-II) unit [15]. A message is considered to be delivered if it is correctly received by any of the Road Side Units (RSUs), which will send it to the control center. RSUs are supposed to be placed in strategical places by entities interested in collecting the information. The location of the RSU is provided to vehicles through a dynamic updating service.

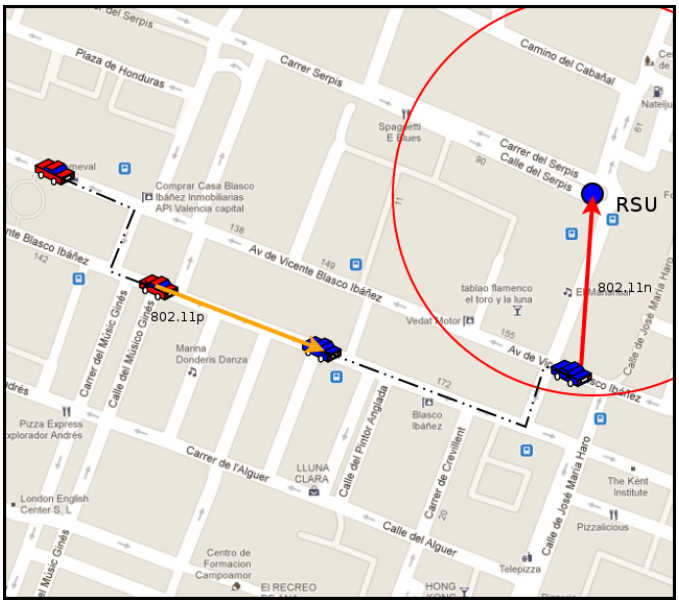

Fig. 1. Typical situation for MSDP, dashed lines represent the movement of the nodes, while solid lines represent wireless transmissions.

We assume that the nodes of our network have an IEEE 802.11n interface for Vehicle to Infraestructure (V2I) communication, and an IEEE 802.11p interface for Vehicle to Vehicle (V2V) communication. Moreover, we also assume that all nodes also have some degree of knowledge about their own route obtained from a Navigation System (NS). That NS can be an interface to different sources like a integrated on board navigation device, or a preloaded static route, like in a train or a bus.

Depending on the reliability of that knowledge, our protocol will assign it a different reliability index that varies from 0 to 1. Considering the fast growing number of on board navigation devices sold worldwide, we strongly believe that the situation previously described can be considered to be true in vehicular networks. Figure 1 shows a typical situation, where dashed lines represent the movement of the nodes, while solid lines represent wireless transmissions.

Briefly explained, our protocol works as follows: Data messages are generated by bundling information from different sensors. Large messages are fragmented into packets that will be stored in the node's buffer. In MSDP a node with packets in its buffer to be transmitted is called *custodian*. *Custodians*

announce their presence and information about the knowledge of their routes to other nodes periodically. We refer to nodes that receive this announcement as *candidates*. *Candidates* will answer to the announcements with a message containing information about their positions and an index depending on their routes. After evaluating this information, the *custodian* will decide, as detailed below, if it is worth forwarding the packets to the best *candidate*, or if it is better to wait for future communication opportunities. It is important to remark that a *custodian* will never remove a packet from its buffer until a *candidate* has been confirmed as the new *custodian*. Finally, when a *custodian* reaches a RSU, it will try to use this communication opportunity to deliver as many packets as possible. Once a RSU has received a packet, the packet will be sent over to the service specific control center inside the backbone. The control center will then reassemble the packets into the original message and process the content.

In an attemp to cope with the problems exposed in previous sections, our proposal properly handles the following issues:

Redundancy: After fragmenting the messages into packets, we add a percentage of redundancy packets using Foward Error Correction (FEC) techniques. It means that, if a message needs N packets, $N*\alpha$ packets will be sent, where redundancy factor $alpha$ is greater than 1 and depends on the configuration. In our scheme all these packets are generated and distributed, but the original message can be reassembled with just N packets. This redundancy allows reducing the impact of possible packet losses.

Dynamic parameters: In MSDP some parameters are determined dynamically using the available information about the road where the car is currently located and its speed. For example, the interval between announcements is determined by the speed of the node and the speed limit of the current road.

Reactive mechanism: Only *custodians* actively announce their presence to other nodes, while *candidates* only send messages as an answer to announcements. This mechanism slightly increases the time required to detect a transmission opportunity, but it strongly reduces both the amount of resources consumed and the channel congestion.

Channel prediction: In our protocol every message contains information about the position and velocity of the source node. Using this information, nodes are able to omit transmissions that would lead to a waste of resources, as occurs when a message is sent to a node close to the maximum transmission range.

A. Routing decision

The main novelty of our proposal is the way our protocol makes routing decisions. In MSDP, *custodians* use the value of a function, called UtilityIndex, to determine which is the best *candidate* to forward each packet, or even whether it is better to keep the packet in the buffer and ignore the transmission opportunity. The UtilityIndex, which depends on three parameters, is calculated locally and communicated to neighbor nodes through MSDP messages. The higher the UtilityIndex is, the better the candidate. It is defined by the following function:

$$UtilityIndex = \frac{P^2}{T} * Q \tag{1}$$

The three parameters, P, T, and Q are defined as follows:

Trustworthy factor (P): This parameter tries to quantify the reliability of the Navigation System (NS), being higher values associated with more reliable NS data.

Time to reach a RSU (T): Using the information obtained through the NS, every *candidate* estimates the time, in seconds, needed to reach the following RSU. Obviously, better *candidates* are associated with lower times to reach a RSU. T is defined by the following equation:

$$T = \frac{log(t+1)}{log(\tau)} \tag{2}$$

being t the time to reach the next RSU expressed in seconds, and τ the maximum delay considered for the application. This function gives more emphasis to small time differences.

Transmission availability (Q): Using the information obtained from the NS, nodes are able to estimate the duration of the next transmission opportunity with a RSU, and also the average transmission rate of nodes connected to that RSU. Therefore, they can estimate the amount of data that they will be able to deliver. Our protocol uses the ratio between the amount of data contained in its buffer and this estimated value (q) to prioritize those nodes with a ratio closer to zero. Q represents this availability, and it is defined by the following function:

$$Q = max[\frac{log(\beta * (1-q))}{log(\beta)}, 0] \tag{3}$$

being q the rate previously mentioned, and β an application parameter that modifies the slope of the logarithmic function. Given an amount of data for the next transmission opportunity, Q decreases when the amount of data in the buffer increases.

B. Data format

In our protocol, nodes generate data messages following the format shown in Figure 2. A data message is defined by a tuple (*SourceID*, *MessageID*, *Timestamp*). The data message is then fragmented into several data packets; the number of packets depends on the size of the original data message. Redundancy information packets will also be generated if necessary. Each data packet includes the original tuple to allow reassembling the message at the destination, along with three new fields, where the first of them indicates the fragment number, the second one indicates the total amount of data packets necessary to reassemble the original data message, and the last one indicates the maximum number of hops that

Fig. 2. MSDP data format.

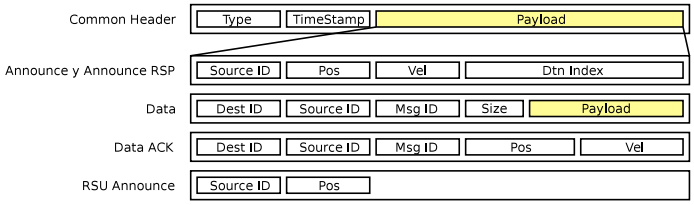

Fig. 3. MSDP message format.

a data packet can traverse. The size of the payload contained in an information packet is fixed.

C. Routing messages

Once data messages are generated and fragmented into packets, packets are individually routed through the network until they are received by a RSU which will send the packets to the control center. With that purpose, our protocol defines 5 types of MSDP messages that are sent using UDP. All of them have a common header that includes both the subtype of the message and the timestamp at which the message was generated. The remaining fields are represented in Figure 3 and described bellow:

- *MSDP Announcement Messages:* This type of message is broadcasted periodically by *custodians*. It contains four additional fields: the unique Id of the node that generated the message, the position of the node, the velocity of the node, and its own UtilityIndex.
- *MSDP Announcement Response Messages:* This message includes the same fields as has MSDP Announcement Messages, but it is generated by *candidates* when an MSDP Announcement is received. They are also sent to the broadcast address.
- *MSDP Data Messages:* This packet contains 4 fields: the source id, the destination id, a local unique message identifier, and the number of data packets serialized in its payload. The encapsulated data packets can belong to different sources.
- *MSDP Data ACK Messages:* This message is used to confirm that an MSDP data message has been received. It contains the information required to identify the confirmed message, as well as about information the position and velocity of the sender.
- *MSDP RSU Announcement:* RSUs announce their position through this message. It contains the position of the RSU and its ID.

D. Nodes Behavior

As we have introduced before, in MSDP there are 3 type of nodes: *custodians*, *candidates*, and RSUs. Below we present the complete behavior of every type of node and the pseudocode for both custodians and candidates:

Custodians:

1) Start announcing their position periodically through MSDP Announcement Messages and then wait for replies from *candidates*.
2) After receiving an MSDP Announcement Response the *candidate* sender ID is stored on a list; then, after a certain delay, a transmission starts with the best of the stored *candidates*.
3) In case an MSDP RSU Announcement is received, a transmission with the RSU will be immediately started.
4) Aiming at reducing the resource consumption, the next MSDP Announcement message scheduled will be omitted if an MSDP Announcement message from another neighbor custodian is received.
5) During a transmission, several data packets are encapsulated inside the payload of MSDP Data messages. The size of MSDP Data messages is limited by network's Maximum Transfer Unit (MTU).
6) When an MSDP Data ACK is received, confirmed data packets are removed from the buffer.

Algorithm 1 Pseudocode of custodians nodes.

```
while packetsInBuffer()
      wait(time)
      sendAnnouncement()
      startTimer(waitTransmissionTime)
end
if receivedRSUAnnouncement(RsuId)
  startTransmission(RsuId)
end
if receivedAnnouncementResponse(candidateId)
  store(candidateId, candidateList)
end
if waitTimerExpired()
  if cadidateList.empty()
    cadidateId = getBestCandidate(candidateList)
    startTransmission(candidateId)
  end
end
if receivedACK(messageId)
  remove(messageId)
end
```

Algorithm 1 shows the pseudocode of custodian nodes.

Candidates:

1) Remain in a passive state until an MSDP Announcement message is received from a *custodian*.
2) After receiving an MSDP Announcement message containing an UtilityIndex lower than its local UtilityIndex, they will send an MSDP Announcement Response message to announce themselves. To avoid collisions, a small random time is introduced before sending the MSDP Announcement Response.
3) To avoid wasting resources, if an Announcement Response containing an UtilityIndex better than the local UtilityIndex is received, and a new MSDP Announcement Response message was scheduled as well, the last one will be omitted.

Algorithm 2 Pseudocode of candidate nodes.

```
if announcementRcvd(custID, custUtilIdx)
  if myUtilIdx > custnUtilIdx
    waitTime()
    sendAnnouncementRsp(myUtilIdx)
  end
end
if announcementeRspRcv(candUtilIdx)
  if candUtilIdx > myUtilIdxANDrspIsScheduled()
    cancelResponse()
  end
end
if dataMsgRcv(msgID, data)
  result = decapsulateData(data)
  if result == OK
    sendAck(msgID)
  end
end
```

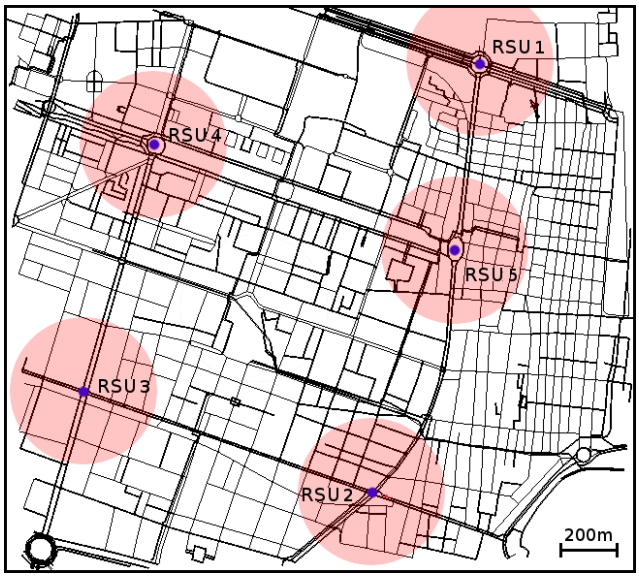

Fig. 4. Map of the city of Valencia used in our simulations.

4) If an MSDP Data message is received, the data packets contained in its payload are decapsulated and stored in the buffer, and so the node becomes a *custodian* node.
5) If the data packets were stored properly, they will be confirmed with an MSDP Data ACK message.

Algorithm 2 shows the pseudocode of candidates nodes.

RSUs:

1) Announce their presence through RSU Announcement messages.
2) When an MSDP Data message is received, the data packets are immediately decapsulated and sent to the control center.
3) If the data packets were sent properly, they will be confirmed with an MSDP Data ACK message.

IV. Simulation Environment

We compared our MSDP protocol against the Epidemic protocol [3] through simulations. The Epidemic protocol provides an upper bound for the delivery probability, and a lower bound for the delay time when enough resources are available. In this section we will detail the tools and the configuration that we have used.

A. Simulation Tools

We have implemented MSDP using the ns3 [16] simulator. Ns3 is an event-driven simulator that includes detailed implementations of the 802.11 physical and MAC layers.

B. Mobility Traces

One of the most important issues in DTN simulations is the mobility pattern. The mobility patterns of our simulation have been generated by a tool designed by our research group called Citymob for Roadmaps (C4R) [17], which is based on the Simulation of Urban MObility (SUMO) [18] tool.

C4R generates pure random routes, meaning that a car starts its route in a random road of a real map, and a destination road is also chosen randomly. Then, the shortest path through the road network is calculated using the Dijkstra algorithm. When a car arrives to its destination, it shuts down all interfaces and is removed from the map. We consider that this behavior is a better approximation to real mobility than other proposed mobility models in which cars never stop, following infinite routes. We want to remark that the duration of the routes is unpredictable, and removing a car means that all the packets contained in its buffer will be lost. In a real situation packets could be reintroduced in the network when the node starts a new route, but only under the assumption that the maximum delay is much lower than the average stop time; we can also assume that all the packets contained in the buffer when a node stops will never arrive. Moreover, as a consequence of the behavior described above, the node density varies throughout the simulation time.

The mobility traces are generated for the city of Valencia, within an area of 2325x2160 meters, represented in Figure 4. We believe that this map is a good example of an average sized European city. In this scenario we located five different RSUs in some strategic avenues of the city.

C. RSU locations

We assumed that RSUs are located by an entity whose objective is to collect as much traffic information as possible. According with this assumption, we placed the RSUs on the avenues of our network with a bigger volume of road traffic. Figure 4 shows the location of the RSUs and their communication ranges.

D. General parameters

In our simulated scenario every node has 2 interfaces: an 802.11n interface tuned at the frequency of 2.4 GHz and used

for V2V communication, and an 802.11p interface tuned at the frequency of 5 GHz and used for V2I communication. UDP parameters, as well as IP, ARP and MAC parameters, take their default values. Both interfaces transmit at the maximum allowed power in Europe: 20 dBm and 33 dBm, respectively.

E. Propagation models

We consider that the use of very simplistic propagation models is one of the main drawbacks of previous studies in this topic. As demonstrated in [19] and [14], this issue must be seriously considered. In an attempt to accurately model real world conditions, we decided to combine the two ray ground propagation model and the Nakagami fading model [13].

F. Generated data traffic

Every node in our network scenario generates a message with a size of 2500 Bytes every 5 seconds. The size of the fragments is 231 Bytes plus headers, making a total size of 256 Bytes. The traffic generation will be stopped after the first 100 seconds of our simulation. The simulation will last 3600 seconds.

G. Different scenarios and repeatability

With the objective of evaluating the impact of incrementing the number of nodes in our network, we have simulated 4 different scenarios with 30, 63, 125, and 188 nodes, varying the initial node density from 6 nodes/km^2 to 37 nodes/km^2. Moreover, to achieve reasonably conclusive results, we have simulated every scenario 30 times varying the mobility traces and the seed of the random number generator. In order to obtain comparable results, our simulations use the same mobility traces for both the MSDP and the Epidemic protocol, i.e. the *ith* iteration with a specific number of nodes uses the same mobility traces for both experiments. Measures are represented with a 95% confidence interval.

V. RESULTS

In this section we present a performance comparison of the MSDP protocol against the Epidemic one. Results show that our proposal performs better than the Epidemic protocol for almost every simulated scenario and under all the selected metrics.

Figure 5 shows the average delivery probability for both protocols at different node densities. At low node densities the Epidemic protocol behaves slightly better than our proposal. This is because the Epidemic protocol ensures the optimal route is used when the available resources, in terms of channel capacity and node mobility, are enough. However, our protocol can miss some multi-hop paths. The maximum delivery probability is obtained with a moderate network density; at this point our protocol performs a 13 % better than the Epidemic protocol. When the number of nodes increases beyond this value the delivery probability decreases. In the case of the Epidemic protocol, it is due to general network congestion; however, in the case of MSDP, it is due to an over-estimation of the throughput towards the RSUs in the presence

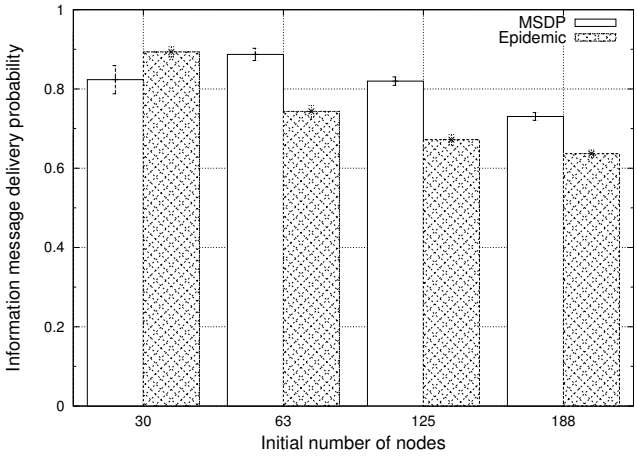

Fig. 5. Delivery probability.

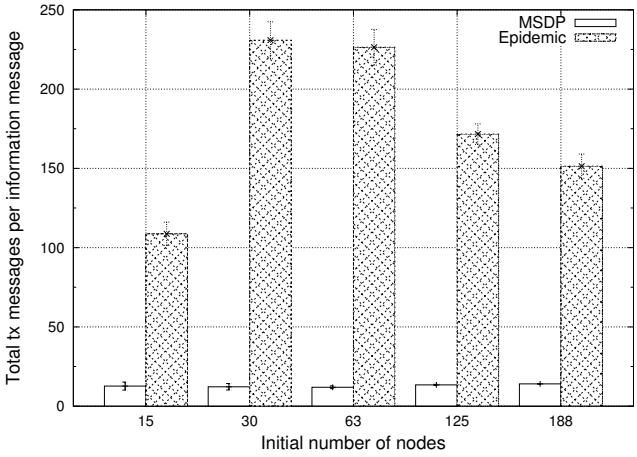

Fig. 6. Total transmissions.

of congestion. Under such conditions the *custodians* are not able to empty their buffers and, as consequence, packets are discarded when the nodes are turned off.

Figure 6 shows the average total number of MSDP message transmissions per data message generated. This metric accounts for every type of message (described in section III) sent to the wireless channel. The Epidemic protocol, even for low densities, generates many more transmissions than our proposal. It is important to notice that, when doubling the initial number of nodes from 15 to 30, the Epidemic protocol doubles the number of transmissions as well, while under our proposal it almost does not increase. At high densities, the high congestion generated by the Epidemic protocol drastically reduces the transmission opportunities between nodes and, as consequence, the total number of transmitted MSDP messages per data message decreases. From our point of view, this metric is very important since it is expected that information collecting applications will coexist with several other applications, and so the imposed overhead will be even higher.

Figure 7 represents the average delay for the received data messages. It shows that, when using our protocol, the mean delay decreases when the amount of nodes in the network

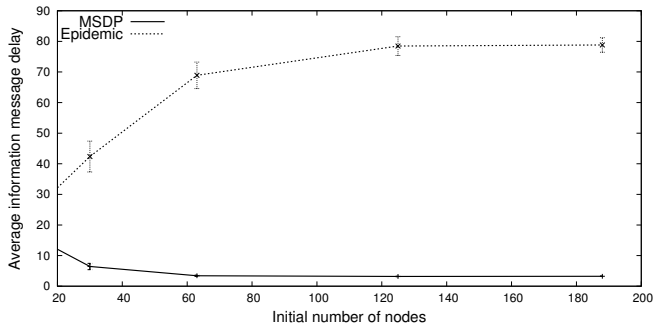

Fig. 7. Average data messages delay.

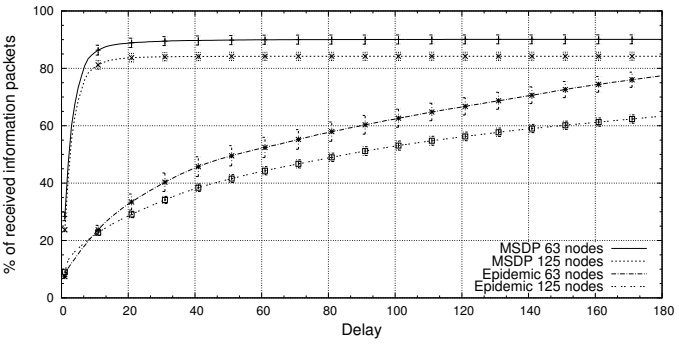

Fig. 8. Cumulative distribution of data packets delay.

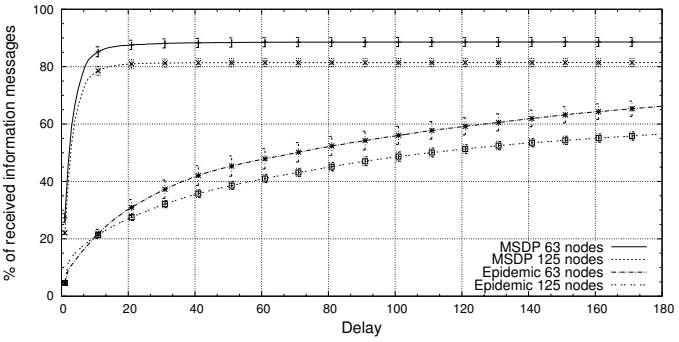

Fig. 9. Cumulative distribution of data message delay.

increases. On the other hand, when using the Epidemic protocol, the mean delay increases when the number of nodes increases. This behavior is a consequence of the huge overhead introduced by the Epidemic protocol, which saturates the network capacity. We have also represented the cumulative distribution of the data messages delay in Figures 8 and 9, which is a better metric to evaluate the complete behavior of a protocol. They show that, when using MSDP, only a small percentage of the received data packets and messages will experiment a delay higher than 15 seconds while, when using the Epidemic protocol, a high number of messages are received more than 1 minute later.

Comparing figures 8 and 9, we can see that the percentage of data messages received is slightly smaller than the percentage of data packets received. This difference becomes bigger in the case of the Epidemic protocol. The cause for this different behavior when comparing our proposal against the

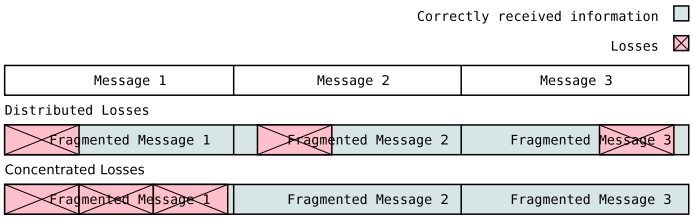

Fig. 10. Distributed and concentrated losses.

Epidemic protocol is the kind of losses that both protocols experiment. In the case of MSDP, losses are produced by invalid routes that are usually taken by the majority of the fragments of a data message. On the other side, the multi-copy scheme of the Epidemic protocol increases the probability of multipath routing for data packets; as a consequence, losses usually affect several data messages. Figure 10 illustrates why distributed packet losses cause more data message losses than concentrated ones. In this example, a 33% of distributed packet losses produce 100% of message losses, while the same percentage of concentrated losses only produces a 33% of message losses.

VI. Conclusions

In this article we proposed MSDP (Map-based Sensor-data Delivery Protocol), a DTN routing protocol that combines information obtained from the Geographic Information Service (GIS) with the actual street/road layout obtained from the Navigation System (NS) to find the best route.

The novelty of our proposal is that MSDP is not simply based on geographic positions or distances, but routing decisions consider the programmed route of the vehicle, the expected time to reach the message's destination, the amount of data stored in the vehicle's buffer, the amount of data expected to be exchanged with the destination, as well as the degree of trust of the source for all the considered information to efficiently deliver sensor data messages to the control center.

Using accurate mobility and propagation models, we thoroughly evaluated our protocol against the well-known Epidemic protocol. The results showed that MSDP achieves a higher delivery probability and a smaller average delay with much lower channel congestion levels.

Acknowledgments

This work was partially supported by the *Ministerio de Ciencia e Innovación*, Spain, under Grant TIN2011-27543-C03-01.

References

[1] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-Tolerant Networking Architecture," no. RFC 4838. IETF, Apr. 2007. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4838.txt

[2] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '03. New York, NY, USA: ACM, 2003, pp. 27–34. [Online]. Available: http://dx.doi.org/10.1145/863955.863960

[3] A. Vahdat and D. Becker, "Epidemic Routing for Partially Connected Ad Hoc Networks," in *Technical Report CS-200006*, Apr. 2000.

[4] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, ser. WDTN '05. New York, NY, USA: ACM, 2005, pp. 252–259.

[5] J. Xue, X. Fan, Y. Cao, J. Fang, and J. Li, "Spray and Wait Routing Based on Average Delivery Probability in Delay Tolerant Network," in *Networks Security, Wireless Communications and Trusted Computing, 2009. NSWCTC '09. International Conference on*, vol. 2. IEEE, Apr. 2009, pp. 500–502.

[6] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," in *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7. New York, NY, USA: ACM, Jul. 2003, pp. 19–20.

[7] T.-K. Huang, C.-K. Lee, and L.-J. Chen, "PRoPHET+: An Adaptive PRoPHET-Based Routing Protocol for Opportunistic Network," in *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*. IEEE, Apr. 2010, pp. 112–119.

[8] R. Frank, E. Giordano, P. Cataldi, and M. Gerla, "TrafRoute: A different approach to routing in vehicular networks," in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2010 IEEE 6th International Conference on*. IEEE, Oct. 2010, pp. 521–528.

[9] Z. Li and H. Shen, "A Direction Based Geographic Routing Scheme for Intermittently Connected Mobile Networks," in *Embedded and Ubiquitous Computing, 2008. EUC '08. IEEE/IFIP International Conference on*, vol. 1. IEEE, Dec. 2008, pp. 359–365.

[10] I. Leontiadis and C. Mascolo, "GeOpps: Geographical Opportunistic Routing for Vehicular Networks," in *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*. IEEE, Jun. 2007, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/WOWMOM.2007.4351688

[11] P. C. Cheng, K. C. Lee, M. Gerla, and J. Härri, "GeoDTN+Nav: Geographic DTN Routing with Navigator Prediction for Urban Vehicular Environments," *Mob. Netw. Appl.*, vol. 15, pp. 61–82, Feb. 2010. [Online]. Available: http://dx.doi.org/10.1007/s11036-009-0181-6

[12] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, ser. MobiCom '00. New York, NY, USA: ACM, 2000, pp. 243–254.

[13] T. S. Rappaport, *Wireless Communications: Principles and Practice (2nd Edition)*, 2nd ed. Prentice Hall PTR, Jan. 2002.

[14] V. Cabrera, F. J. Ros, and P. M. Ruiz, "Simulation-Based Study of Common Issues in VANET Routing Protocols," in *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*. IEEE, Apr. 2009, pp. 1–5.

[15] International Organization for Standardization, "ISO 15765: Road vehicles, Diagnostics on Controller Area Networks (CAN)," 2004.

[16] "Ns3 website," http://www.nsnam.org/, December 2011.

[17] M. Fogue, P. Garrido, F. J. Martinez, J.-C. Cano, C. T. Calafate, and P. Manzoni, "Using roadmap profiling to enhance the warning message dissemination in vehicular environments," in *36th IEEE Conference on Local Computer Networks (LCN 2011)*, Bonn, Germany, October 2011.

[18] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo - simulation of urban mobility: An overview," in *SIMUL 2011, The Third International Conference on Advances in System Simulation*, Barcelona, Spain, October 2011, pp. 63–68.

[19] M. T. Moreno, S. Corroy, F. S. Eisenlohr, and H. Hartenstein, "IEEE 802.11-based one-hop broadcast communications: understanding transmission success and failure under different radio propagation environments," in *Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems*, ser. MSWiM '06. New York, NY, USA: ACM, 2006, pp. 68–77.