Acknowledgment-based Broadcast Protocol for Reliable and Efficient Data Dissemination in Vehicular Ad-hoc Networks

Francisco J. Ros, Pedro M. Ruiz, *Member, IEEE,* and Ivan Stojmenovic, *Fellow, IEEE*

***Abstract*—We propose a broadcast algorithm suitable for a wide range of vehicular scenarios, which only employs local information acquired via periodic beacon messages, containing acknowledgments of the circulated broadcast messages. Each vehicle decides whether it belongs to a *connected dominating set* (CDS). Vehicles in the CDS use a shorter waiting period before possible retransmission. At timeout expiration, a vehicle retransmits if it is aware of at least one neighbor in need of the message. To address intermittent connectivity and appearance of new neighbors, the evaluation timer can be restarted. Our algorithm resolves propagation at road intersections without any need to even recognize intersections. It is inherently adaptable to different mobility regimes, without the need to classify network or vehicle speeds. In a thorough simulation-based performance evaluation, our algorithm is shown to provide higher reliability and message efficiency than existing approaches for non-safety applications.**

***Index Terms*—Vehicular networks, data dissemination, broadcasting, distributed algorithm, performance evaluation.**

I. Introduction and Motivation

Vehicular Ad-hoc Networks (VANET) consist of collections of vehicles equipped with wireless communication capabilities. Vehicles cooperate to deliver data messages through multi-hop paths, without the need of centralized administration. To achieve this, communication protocols must cope with the mobility of vehicles and the dynamics of wireless signals. Vehicle movements are restricted by the streets layout, traffic signals and other vehicles' movements. This leads to highly partitioned networks with non-uniform distribution of nodes. Furthermore, different scenarios (e.g. highway vs. urban) need to be addressed when studying vehicular networks. Technical challenges in this environment are discussed in [1].

Broadcasting is the task of sending a message from a source node to all other nodes in the network. It is frequently referred to also as *data dissemination*. The design of reliable and efficient broadcast protocols is a key enabler for the successful deployment of vehicular communication services. Most of the envisioned services (ranging from safety applications to traffic management and infotainment [2]) rely on the delivery of broadcast messages to the vehicles inside a certain area of interest. This operation is therefore also known as *geocasting*. For example, location-based services (gas prices, tourist points of interest, and so on) can be advertised from Road Side Units (RSUs) deployed along roads, and the messages would be propagated to the near-by vehicles. Another application is for the Internet-gateway discovery. Vehicle-to-vehicle (V2V) communication system can also be used as a distributed platform for "opportunistic cooperation" among people with shared interests or goals [3]. In [3], authors discuss the concept of a virtual "flea market" over a VANET called FleaNet. In FleaNet, customers, either mobile (i.e., in vehicles) or stationary (e.g., roadside shop owners), express their demands/offers, e.g., their desire to buy or sell an item, via radio queries. These queries are opportunistically disseminated, by exploiting the mobility of other customers, in order to find a customer/vendor with matching needs/resources.

In this paper, we focus on the problem of broadcasting in VANETs without infrastructure support. In some applications, messages originate at infrastructure and V2V communication proceeds from there. Regardless of criticality, the infrastructure for support of data dissemination is currently sparse, and therefore a protocol not relying on it is mandatory building block, as the best effort. Any existing infrastructure could be easily integrated into a given solution for enhanced performance.

Our primary goal is to achieve high reliability while minimizing the total number of retransmissions. In some safety applications, the delivery latency is critical. However, considering all three goals appears to be a very challenging task, and we concentrate here on non-safety applications like the ones previously described. At the same time, vehicle still may not delay retransmission for too long as the reliability would otherwise suffer.

Topology changes due to mobility cause frequent and temporary disconnections. Message might require to be buffered and carried by a given vehicle until a new forwarding opportunity emerges. Several broadcasting protocols have been previously proposed. However, they are designed for either rectilinear highways/roads ([4][5][6]) or urban grids ([7][8]). More surprisingly, only one of them [6] addresses the issue of temporary disconnections in VANET, which is one of its most salient properties.

We develop the *Acknowledged Broadcast from Static to highly Mobile* (ABSM) protocol [9], a fully-distributed adaptive algorithm suitable for VANETs with all mobility scenarios. ABSM automatically adjusts its behavior without keeping track of the degree of mobility sensed by the vehicle. Each node independently decides whether or not to forward a

---

F. J. Ros and P. M. Ruiz are with the Department of Information and Communications Engineering, University of Murcia, Murcia, Spain. E-mail: `{fjros,pedrom}@um.es`.

I. Stojmenovic is with SITE, University of Ottawa, Ontario, Canada, and the Department of Electronic, Energetics and Telecommunications, FTN, University of Novi Sad, Serbia. E-mail: `stojmenovic@gmail.com`.

Digital Object Indentifier 10.1109/TMC.2010.253

received broadcast message. Such decision is solely based on the local information that vehicles acquire from their neighborhood by means of periodic beacon messages. This guarantees ultimate scalability regardless the size of the VANET. The set of parameters in ABSM is minimal and consists only of few natural choices.

In ABSM, a car that receives a broadcast message does not retransmit it immediately. Instead, the vehicle waits to check if retransmissions from other neighbors already cover its whole neighborhood, making its transmission then redundant. To acquire 1-hop neighborhood position information, periodic beacons contain the position of the sender. Such information suffices to compute a *connected dominating set* (CDS). Nodes in the CDS select a shorter waiting timeout than regular nodes. This allows them to retransmit first if their neighborhood has not been already covered. That is, we combine two different techniques, CDS and *neighbor elimination scheme* (NES) [10][11]. Beacons also include identifiers of the recently received broadcast messages, which serve as acknowledgments of reception. This way, nodes can check whether all their neighbors successfully received a message. If this is not the case, a retransmission is scheduled (upon the expiry of a timeout duration). Otherwise, retransmission would be redundant. In both cases, when a new neighbor emerges, nodes restart their evaluation timeout if the message being disseminated is not acknowledged. If the message identifier is actually included within the beacon, the neighbor already got the message and no retransmission is scheduled. Hence, the use of acknowledgments makes the protocol more robust to transmission failures while, at the same time, saves redundant retransmissions.

Temporary disconnection incurs delivery delay to any protocol. Although the described protocol inherently uses the store-carry-forward paradigm, ABSM does not incurs large delivery latencies. Nodes connected to the source will receive the message with small delay, due to propagation via CDS.

On our thorough simulation study, we assess the performance of ABSM and compare it with other competing algorithms. We focus on VANETs in both highway and urban scenarios. Vehicles' movements are generated with a microscopic road traffic simulation package, in order to mimic common scenarios of real vehicular networks. Different mobility conditions are simulated. Under realistic IEEE 802.11p [12] models, ABSM is shown to outperform remaining approaches.

The remainder of this paper is organized as follows. Section II presents literature review. In Section III, we give insight onto the proposed algorithm and describe it in detail. The simulation-based performance evaluation and related discussions are presented in Sections IV–VIII. Finally, Section IX concludes the paper.

II. LITERATURE REVIEW

There exist a plethora of proposed broadcasting protocols for wireless ad-hoc networks. Several surveys describe many of them, such as [13]. Here, we only refer to those techniques and protocols that are directly related to our approach. We also review the broadcasting algorithms that have been specifically designed for vehicular networks (except those with minimizing delay as the primary objective, in critical safety applications).

A. CDS-based broadcasting

The problem of designing efficient broadcast protocols for ad-hoc networks has been deeply investigated for several years. Probably the most common technique to reduce redundant transmissions in a broadcasting task is the use of *connected dominating sets* (CDS). Let $G(V, E)$ be the graph induced by the network topology, so that V is the set of nodes in the network and E represents the connectivity between them. Then, a subset $V_D \subseteq V$ is said to be dominating if each node in V either belongs to V_D or has at least one neighbor which belongs to V_D. V_D is a CDS if it is connected. In CDS-based broadcasting, only those nodes belonging to the CDS are needed to retransmit the broadcast message, and it will indeed reach the whole network. Therefore, the fewer number of nodes in the CDS, the less redundant the broadcast protocol will be.

Unfortunately, the problem of finding the minimum CDS was shown to be NP-hard [14], and many heuristics have been proposed since then. Wu *et al.* described several lightweight backbone construction schemes. We will use a modified definition from [10][11] of the basic concept in [15], because of its reduced message overhead.

Assume that each node x is identified by a unique key $key(x)$. Then, a node is said to be an *intermediate node* if it has two unconnected neighbors [15]. A node u is covered by neighboring node v if each neighbor of u is also neighbor of v, and $key(u) < key(v)$. A node u is covered by two connected neighboring nodes v and w if each neighbor of u is also a neighbor of either v or w (or both), $key(u) < key(v)$, and $key(u) < key(w)$. An intermediate node not covered by any neighbor becomes an *inter-gateway node*. An inter-gateway node not covered by any pair of connected neighboring nodes becomes a *gateway node*. A set of gateway nodes form a CDS.

Wu's concepts require either 1-hop knowledge of neighbors with their position, or 2-hop neighbor topology information. Such information is obtained by means of periodic 'hello' (beacon) messages exchange. Experimental data from several sources confirm that Wu's concepts provide small size CDS on average. Each node makes decisions about CDS membership (in Wu's concept) without communications between nodes beyond the message exchanges that nodes use to discover each other and establish neighborhood information.

A framework and general algorithm in [10][11] is based on two concepts: CDS as the particular type of backbone that provides reliability, and neighbor elimination scheme (NES). In NES [10][11][16], a node does not need to rebroadcast a message if all its neighbors are believed to be covered by previous transmissions. After each received copy of the same message, a node eliminates, from its rebroadcast list, the neighbors that are assumed to have received the same message (based on local knowledge). If the list becomes empty before the node decides to rebroadcast, the retransmission is canceled.

The general Dominating Set and Neighbor Elimination Scheme (DS-NES) [10][11] for intelligent flooding proceeds as follows. The source node transmits the packet. Nodes not in the CDS do not retransmit the packet. Upon receiving the first copy of the message, a node in the CDS will select a timeout period to wait. It will also eliminate from its forwarding list (originally containing all 1-hop neighbors) all neighbors that received the same copy of the message. While waiting, more copies of the packet could be received. For each of them, all neighbors receiving it are eliminated from the forwarding list. When the timeout expires, the node will retransmit if its forwarding list is non-empty, otherwise it will cancel retransmission. This framework was applied in [10][11] using clustering based and Wu's concept based backbones.

The *parameterless broadcast in static to highly mobile* (PBSM) ad hoc networks protocol [17] makes use of the DS-NES framework to develop an adaptive algorithm which does not depend on any parameter or threshold value. Because of its flexibility and good performance, it is used as the basis of ABSM for vehicular ad hoc networks. In PBSM, each vehicle s maintains two lists of neighboring cars with respect to the message being disseminated and local 1-hop knowledge: R and N, containing neighbors that already received (did not receive, respectively) the message. After a delay timeout, s retransmits the message if the list N is nonempty. Both lists R and N are updated with every copy of message and beacon exchange received, which may trigger further retransmissions if N becomes non-empty again. Nodes in the CDS set shorter waiting timeouts than nodes that are not part of it.

B. VANET-specific broadcasting

We limit our review to protocols designed primarily for non-safety applications (and therefore not emphasizing minimal delay as the main objective). Vehicles tend to travel forming groups in highly disconnected networks. Vehicular density can be extremely high in a traffic jam, while surrounding streets or lanes could have low traffic density. This uneven node (and speed) distribution is characteristic of vehicular settings. Therefore, several broadcast protocols specifically designed for such networks have been proposed so far.

A few simple geocasting algorithms are offered in [3]. In the first one, each node periodically broadcasts its query to neighboring nodes. Query is dispersed via mobility and only to 1-hop neighbors. It is then extended toward m-hop retransmission similarly (with decreasing hop counter until reaching 0). In the next scheme, each receiving car will retransmit with certain fixed probability. Further scheme is random walk to spread the query to k proxy cars, and then these cars periodically advertise to inform their 1-hop neighbors. In neighbor split scheme, originator splits k proxy advertisers equally among its neighboring nodes. This continues recursively and then 1-hop neighbors are informed periodically. These schemes do not meet satisfactory reliability objective.

In [4], two solutions which consider vehicles located in one (or few parallel) lanes on a highway, all driving in the same direction, are presented. In the first proposal (sender oriented), the car transmitting the message decides the next forwarder by including the identifier of its farthest neighbor (in the direction of the broadcast propagation) within the message. This approach is not reliable because the intended neighbor might not be reachable when the transmission takes place, since the connectivity was established at a previous beacon message exchange. Such situation would stop the flooding process prematurely. In the second solution, the next forwarder selection is performed at the receiver. The transmitting vehicle appends its own location to the broadcast message. Receivers defer the retransmission for a back-off time which is inversely proportional to their distance from the previous forwarder. In a one-lane highway scenario, the next forwarder is normally the farthest car from the previous forwarder, among those that received the retransmission. This protocol is not intended to guarantee delivery to all the nodes. It only discusses progress between two intersections, which is more precisely a small-scale routing task, and not how to retransmit and provide message to nodes between two forwarders. It also is one-dimensional, and messages may 'jump' over intersections. A variant of this scheme has been proposed to implement cooperative collision avoidance (CCA) [5]. A 1D broadcasting algorithm to disseminate the same message to all cars on a road segment is described in [18]. As in [4], the farthest node from the sender retransmits the message for fast progress. The extension is that the node closest to the middle between two senders retransmits for increased reliability. It is not clear how many such iterations are needed, and how this is extended to 2-dimensional scenarios.

Other variants rely on the MAC layer to improve the broadcasting task in vehicular networks. The *urban multi-hop broadcast* (UMB) protocol [7] is an 802.11-based solution targeted at reducing the broadcast storm and hidden node problems while maximizing the reliability. The broadcast storm is minimized by only allowing the farthest vehicle which receives a message to forward it. For this, after successfully receiving a message, vehicles issue a black-burst jamming signal whose duration is directly proportional to the distance between the transmitter and the receiver. When the signal transmission ends, the vehicle listens to the medium to check if other neighbors are still transmitting a black-burst. If not, that vehicle is the farthest one from the transmitter and forwards the message. The hidden node problem is addressed by adding a request-to-broadcast (RTB) / clear-to-broadcast (CTB) exchange, similar to the case of unicast messages. In addition, reliability is expected to be improved via acknowledgment messages (ACKs, also like unicast). The protocol is designed for dense urban scenarios, with intersections and streets in several directions. Along each street, directional broadcasts take place in the direction of the message propagation. UMB assumes that a repeater is deployed at each intersection, thus initiating directional broadcasts along each of the converging streets. There is also a version of the protocol which substitutes repeaters for regular vehicles which are crossing the intersection [8], therefore eliminating the need of infrastructure.

A highway probabilistic flooding algorithm is proposed in [19]. Front and back counters are updated for received message copies. Before possible retransmission, there is a waiting time that includes the urgency of the message. Probabilistic retransmission decision favors large difference in counters and low sum. Upon retransmission, a node sets another waiting time.

Cluster merging balances counters and reduces retransmission probability. The protocol assumes one direction traffic only, is probabilistic, and has slow merging when one counter is already large.

Three probabilistic and timer based broadcasting suppression techniques for well-connected vehicular networks were proposed in [20]. Their objective is to minimize the well-known broadcast storm problem. In the *weighted p-persistence* scheme, upon receiving a packet, node j waits for a constant time W to receive other potential copies of the message. Let i be the closest neighbor from which the packet has been received. Then, j rebroadcasts the packet with probability $p_{ij} = D_{ij}/R$ if it receives the packet for the first time, and discards it otherwise, where D_{ij} is the distance between i and j and R is the transmission radius. In case j decides to not retransmit, it waits for an additional time δ (accounts for transmission and propagation delays) to overhear the same message again from any neighbor. If this is not the case, j rebroadcasts with probability 1. In the *slotted 1-persistence* scheme, j selects time slot $S_{ij} = N_s(1 - \lceil D_{ij}/R \rceil)$, where N_s is the maximum number of slots. It rebroadcasts (with probability 1) at the assigned slot if it receives the packet for the first time and does not hear any duplicate before the assigned slot; otherwise the packet is discarded. Finally, in the *slotted p-persistence* scheme, rebroadcasting is done with pre-determined probability p instead of probability 1, and retransmission with probability 1 is scheduled if no duplicate was heard within certain time limit. Versions of the algorithms using the received signal strength (RSS), instead of position information, are also described.

The solutions described so far are designed for either highway ([4][5][20]) or urban ([7][8]) scenarios. More surprisingly, none of them addresses properly the issue of temporary disconnections in VANET, which is one of its most salient properties. The Distributed Vehicular Broadcast (DV-CAST) protocol [6] is the only solution we found in the literature that explicitly addresses the various connectivity conditions which are present in vehicular networks, although it can only be applied to rectilinear streets with several lanes (like highways). Vehicle behavior is decided by its status. It is in *well-connected* status if it has at least one neighbor of the same cluster in the message forwarding direction. In such case, the well-connected vehicle runs one of the broadcast suppression techniques described in [20]. A vehicle is operating in sparsely-connected regime if it is the last one in a cluster of vehicles. In addition, it is said to be in a *sparsely-connected neighborhood* if it has at least one neighbor in the opposite direction. Otherwise, the vehicle is in a *totally disconnected neighborhood*. Upon receiving a packet, the sparsely-connected vehicle immediately rebroadcasts it. If it moves in the same direction as the original message source, the packet is then discarded. Otherwise the packet is carried until it expires or can be retransmitted back to the original message forwarding direction. Message is carried afterwards until an implicit acknowledgment is received (from another vehicle with greater hop count), and is being retransmitted in the meanwhile if new neighbors are identified. Vehicle in totally disconnected mode carries the message until a new neighbor is identified, retransmits it with probability 1 immediately, and discards it afterwards.

There are a few drawbacks in the DV-CAST protocol. The algorithm is tight to a highway structure of the traffic. The notions of neighbor in "message forwarding" and in "opposite direction" may often be unclear, e.g. for city scenarios with several roads joined at an intersection. Therefore, DV-CAST will not work in such scenarios. Further, the algorithm also depends on whether or not a sparsely connected vehicle moves in the same direction as the original message source. However, there are scenarios where the message source is static. Next, after a node rebroadcasts the packet in totally disconnected mode, it deletes it and therefore the next coming neighbors will not receive this message in a scenario where all vehicles on the road are totally disconnected. Finally, after each transmission, neighboring vehicle is assumed to have received it, and there are no attempts to guarantee delivery to all cars in an area.

III. THE ABSM PROTOCOL

A. Overview

We propose an adaptive broadcast protocol, ABSM, that is suitable for a wide range of mobility conditions. The main problem that a broadcast protocol must face is its adaptability to the very different vehicular arrangements in real scenarios. It should achieve high coverage of the network at the expense of as few transmissions as possible, regardless on whether the network is extremely dense (e.g. big cities at rush hours) or highly disconnected (e.g. highways at night).

ABSM is localized, and based on applying the CDS and NES concepts on the currently available neighborhood information. Our protocol is not tied to a given CDS heuristic, so that any of them can be employed. In addition, ABSM assumes ideal communication radios to estimate the network connectivity and therefore apply the CDS/NES techniques. Since real communication links are far from ideal, the protocol makes use of broadcast acknowledgments to assure the reception of the message or retransmit it. A message is acknowledged during its whole lifetime. At expiration, it is removed from the vehicle's buffer and no more acknowledgments are issued. Given that broadcast messages are acknowledged, it is assumed that they can be uniquely identified.

Vehicles are assumed to be equipped with Global Positioning System (GPS) receivers. Periodic beacon messages are exchanged to update the vehicles' local topology knowledge. The position of the sender is included within the beacons, which suffices to calculate a CDS backbone after each beacon message round. The source node transmits the message. Upon receiving the message for the first time, each vehicle initializes two lists: list R containing all nodes believed to have received the message (according to local knowledge gained via beacons), and list N containing those neighbors in need of the message. Then, each receiving node sets a timeout waiting period. If a node is not in the CDS then it selects longer timeout than the nodes from the CDS, so that the latter react first. For each further message copy received, and its own message sent, every node updates R, N and the timeout. At the end of the timeout period it transmits if N is

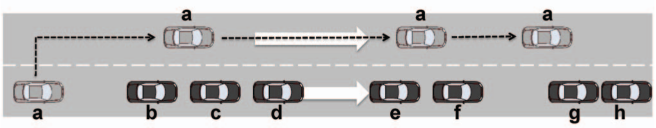


Fig. 1. Common vehicular scenario. Vehicle a overtakes vehicles $b-f$.

nonempty. Both ways, the message is buffered until it expires. For each beacon message received, N and R are updated according to the presence or absence of acknowledgment. Nodes that are no longer 1-hop neighbors are eliminated from these lists. Regardless of previous decisions, all nodes that so far received the broadcast message check whether N becomes non-empty. If so, they start a fresh timeout. In addition, acknowledgments of received broadcast messages are piggybacked to periodic beacons. Nodes that were included in R because they were believed to have received the message, but did not actually get it, are later removed from R and inserted into N. This algorithm is executed for each different message. Therefore, the beacon size increases linearly with the number of simultaneous broadcasting tasks.

We will illustrate the protocol behavior on one example. Given the scenario depicted in Figure 1, vehicle a generates a broadcast message which is first buffered by a, and then received by b, c, d. Receivers set up a waiting timeout which is shorter if the vehicle belongs to the computed CDS. Let d be in the CDS, thus it retransmits first. Vehicles b and c cancel their retransmission because all their neighbors have been covered by d's forwarding. Vehicles e and f receive the message. However, none of them have uncovered neighbors, so the retransmission does not take place.

Vehicle a speeds up and overtakes vehicles $b-f$. In the case of PBSM, new transmissions would occur because new neighbors e and f must be covered by a (and vice versa). However, they are redundant because all the vehicles already received the message. ABSM saves these redundant transmissions because the beacons contain the acknowledgment of the message, and therefore the newly discovered neighbors are not covered again.

B. Protocol details

Pseudo-code of ABSM is given in Algorithm 1. Upon receiving the broadcast message, vehicle x includes in R the sender and all its known neighbors (and starts to_ack timers), because they have likely also received the message (lines 4-10). Accordingly, those vehicles are removed from N (6,9). The remaining neighbors of x which are not connected to the sender (their distance is greater than transmission radius r) are inserted into N (12-13). There exists a timeout function to_ev which assigns a waiting time to each vehicle before its possible retransmission. to_ev is proportional to $1/|N|$, where $|N|$ is the number of elements in N, and depends on whether or not the node is currently in the CDS (shorter waiting time if in the CDS). The rationale is to provide vehicles that have more neighbors in need of the message priority to retransmit first. If several neighbors have the same status and number of neighbors in need of the message, they will obtain the same to_ev value. However this does not mean an increased number of collisions, since ABSM runs at the network layer and these messages still have to contend to access the medium at the link layer (IEEE 802.11p).

Whenever a new neighbor (except the source of a newly received message) is inserted into R, x (vehicle under consideration) initializes a timeout to_ack attached to such neighbor (10). It is used to wait for the acknowledgment of reception. We set to_ack to approximately the *beacon holding time* which is the maximum amount of time a node waits without receiving beacons from a neighbor before deleting it from its neighbor list (43-47). This allows nodes to still receive acknowledgments after more than one beacon interval in the case the original message was not initially received but later it was received from other retransmitters. That is, it allows saving some extra retransmissions by just waiting a bit longer for those acknowledgments. If to_ack expires and the acknowledgment has not been received, the corresponding neighbor is moved from R to N (39,42), or it is removed from the lists if its expected beacons were not received. If N was empty and a new element is inserted, to_ev is reactivated if it was not already running (40-41). In case to_ev was running, it is updated according to the new value of $|N|$ and the elapsed time since the last schedule (40-41). In case N becomes empty ($|N| = 0$), x cancels to_ev and decides not to retransmit (17-18). When to_ev expires, if N is not empty then x retransmits the message and moves the content of N to R (causing the activation of timeouts to_ack) (31-37). For each acknowledged message listed within a beacon from neighbor b, x cancels the associated to_ack (23-24) and adds/confirms b in R (removing it from N if it was there) (25). Note that some acknowledgments can be received before the message itself, so R may be nonempty already when the message is received for the first time.

C. Discussion

ABSM is an appropriate solution for VANETs. First, the protocol is scalable because it only needs local information to perform the broadcasting task. Local information is obtained from beacon messages. This does not increase message overhead because they are needed by safety applications and are mandated by on-going standards like DSRC [21]. The only additional overhead comes from the inclusion of the acknowledgments inside periodic beacons, since the sender's position is included by default. Acknowledgments appear the best strategy to guarantee delivery to all vehicles. Receivers may malfunction, and physical layer modeling has large randomness component even if made with accurate parameters.

In order to minimize the number of message transmissions while preserving reliability, ABSM creates a broadcast delivery backbone based on a CDS heuristic. Vehicles in the CDS choose a shorter timeout, to give them higher priority to retransmit. In addition, NES is employed to further reduce the number of redundant transmissions. This approach is appropriate for vehicular scenarios such as urban layouts with intersections. Vehicles located at junctions which are the only ones with connectivity with other vehicles at converging streets,

Algorithm 1: Pseudo-code of ABSM.

```
Initialize
$B \leftarrow$ neighbor set of this node; $r \leftarrow$ comm. range;
$R \leftarrow \emptyset$; $N \leftarrow \emptyset$;

Event message copy received from neighbor $s$ or generated by this node $s$
Insert message id in subsequent beacons;
$R \leftarrow R \cup \{s\}$; $N \leftarrow N \setminus \{s\}$;
foreach $n \in B$ do
    if $dist(n, s) \leq r$ then
        $R \leftarrow R \cup \{n\}$; $N \leftarrow N \setminus \{n\}$;
        schedule $to\_ack$ for $n$;
    else
        if $n \notin R$ then
            $N \leftarrow N \cup \{n\}$;
if $s = source$ then
    forward message via 802.11;
else
    if $N = \emptyset$ then
        cancel $to\_ev$;
    else
        schedule $to\_ev$;

Event beacon received from neighbor $n$
Add $n$ to neighbor set; Compute CDS;
if beacon contains ack then
    cancel $to\_ack$ for $n$;
    $R \leftarrow R \cup \{n\}$; $N \leftarrow N \setminus \{n\}$;
else
    if $n \notin R$ then
        if $n \notin N$ then
            schedule $to\_ev$;
        $N \leftarrow N \cup \{n\}$;

Event $to\_ev$ expires
if $N \neq \emptyset$ then
    $R \leftarrow R \cup N$;
    foreach $n \in N$ do
        schedule $to\_ack$ for $n$;
    $N \leftarrow \emptyset$;
    forward message via 802.11;

Event $to\_ack$ expires for neighbor $n$ and ack from $n$ never received
$R \leftarrow R \setminus \{n\}$;
if $n \notin N$ then
    schedule $to\_ev$;
$N \leftarrow N \cup \{n\}$;

Event beacon from $n$ not received for last $beacon\_hold\_time$
if $N = \{n\}$ then
    cancel $to\_ev$;
$N \leftarrow N \setminus \{n\}$;
Remove $n$ from neighbor set; Compute CDS;
```

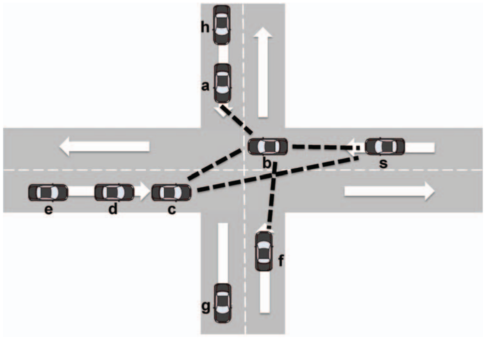


Fig. 2. Intersections in vehicular scenarios. Dotted lines represent connectivity between the subset of vehicles surrounding the intersection. Vehicle s initiates the broadcasting task. In A/PBSM, b is used as relay and therefore the message propagates for every converging street. Forwarding progress-based approaches would select c as relay, and the message would only propagate through the current street.

will be selected as dominating and therefore will retransmit sooner to propagate the broadcast message along those streets (see Figure 2). This is achieved by means of the own CDS selection mechanism, without ever dealing directly with the notion of 'intersection' in the protocol description. Note that VANET-specific protocols (including those designed for safety applications) in which the forwarder selection is based on the concept of progress from the transmitter [4][5][20], fail to support this scenario. In Figure 2, if vehicle c receives s's transmission and forwards first (since it is farther from s than b), vehicles a, f located at converging streets would not receive the message. Other approaches [7][8] need to explicitly handle the case of intersections by starting new directional broadcasts.

In the *unit disk graph* (UDG) model, two nodes u, v are neighbors and can therefore directly communicate iff $distance(u, v) \leq r$, where r is the radius of the communication range. We demonstrate that CDS concept used here is effective in VANET. Actual CDS definition in realistic physical layer is complicated because physics is complicated: the link between any two cars is probabilistic so it is not even clear when to declare them neighbors. CDS was indeed here defined using UDG as approximation, but then we show that such use of simple approximated CDS is just enough for satisfactory performance of ABSM under realistic VANET physics. Computing a CDS in a VANET environment comes for free since beacons with geographic information are periodically triggered. The use of acknowledgments makes the protocol more suitable to the VANET fading environment. If a message is not received by a theoretical neighbor, the latter does not announce its reception in subsequent beacons and the vehicles with the message will issue new transmission. If the message is received by a theoretical non-neighbor, there will be no retransmission later if that node suddenly becomes a neighbor. The superiority of ABSM over PBSM is explained by this correction of UDG-based initial estimate. PBSM updates lists R and N implicitly assuming the UDG model. The inclusion of acknowledgments in ABSM protects the protocol against this assumption, since message losses are expected to happen. This allows ABSM to perform well in real environments.

Finally, and contrary to protocols like DV-CAST, our solution does not need to determine the traffic regime that is sensed by the vehicle. This simplicity is a great advantage: since there are no different internal states, flaws due to unexpected situations are less prone to appear. Nowhere in the ABSM protocol it matters what is speed of vehicle or if vehicle is at an intersection. It therefore provides smooth adaptation to network dynamics including intersections, without changing its behavior. For comparison, GPCR protocol [22] changes when vehicle is at intersection. Also, determining which nodes are located at intersections requires downloading maps in addition to position information.

IV. EVALUATION SET-UP

We have performed different tests to assess the performance of ABSM. The simulation work has been done with *The Network Simulator ns-2*[1], version 2.33. Along with ABSM, we also implemented competing algorithm DV-CAST and two variants of PBSM: PBSM-2t, which uses 2-hop topology information as described in [17]; and PBSM-1p, employing 1-hop position information. PBSM-1p, PBSM-2t and ABSM implement the CDS heuristic described in [10][11]. We have used vehicles' unique identifiers as keys. In all PBSM variants, the timeout to_ev is computed as in Eq. 1, while to_ack is fixed to a constant value in ABSM. The effect of parameters W and to_ack are studied in Section VI.

$$to\_ev = \begin{cases} \frac{W}{|N|}, & \text{if in CDS} \\ W \cdot \left(1 + \frac{1}{|N|}\right), & \text{otherwise.} \end{cases} \quad (1)$$

For DV-CAST, our implementation employs the weighted p-persistence algorithm as the broadcast suppression technique, with parameters $W = 0.25sec$ and $\delta = \frac{W}{10}sec$ (see Section II-B). The other slot-based approaches were not chosen because, as recognized by the authors, they depend on parameters which may be hard to tune in practice [20]. To determine the vehicle status, DV-CAST uses concepts such as the message forwarding direction, the position inside a cluster of vehicles, and the presence of neighbors in the same or opposite direction [6]. The position of the sender and its direction is included in periodic beacons. In addition, data packets are augmented with a network header that indicates the position and direction of the source, as well as the position of the last forwarder (previous hop). Such information suffices to derive the status of each vehicle.

We consider specific vehicular scenarios and movements for two different setups, namely highway and suburban. The former consists of a $4km$ long rectilinear highway with two lanes per direction, while for the latter we have employed a square grid of $4km^2$ with two crossing streets that converge at the center of the square. Each street has two lanes in opposite directions. Vehicles must stop at intersections when others are crossing, so that traffic jams are longer here than in the highway setup. DV-CAST has not been included in this set of simulations because it is not designed for urban scenarios with intersections (as confirmed in [9]).

TABLE I
SIMULATION PARAMETERS FOR THE VEHICULAR SCENARIOS.

Simulation Time	120 sec (after steady state)
Traffic Rate	(1/75, 1/60, 1/45, 1/30, 1/15, 1/5) veh/sec/route
Maximum Speeds	(50, 80) km/h
Beacon Interval	0.5 sec
Beacon Hold Time	1.5 sec
W	(0.1, 0.25, 0.5, 1, 2) sec
to_ack	(0.6, 1.1, 1.6, 2.1) sec
Transmission Power	1.52 mW
Carrier Sense Thres.	802.11p: -94 dBm
Contention Window	802.11p: [15, 1023]

In order to create highway and suburban scenarios, as well as to generate the mobility traces of the vehicles, we have employed the *SUMO* microscopic road traffic simulation package[2]. This allows us to simulate common vehicular situations such as overtakes and stops at intersections. This leads to intermittent connectivity and uneven distribution of vehicles. In each scenario, we defined several routes which are followed by the vehicles. SUMO injects cars in each route according to a given traffic rate, measured in injected vehicles per second. In order to get a wide range of network connectivity, we have varied the traffic injection rate per route from $1/75$ to $1/5$ vehicles per second. The higher the traffic injection rate, the higher the network density. In the suburban scenario we defined more routes, so that a lower rate $(1/15)$ generates a network density comparable to the highway setup with higher rate $(1/5)$. Some figures and tables in this section are labeled with the reciprocal of this rate, i.e., with the interval between the injection of consecutive vehicles (from 75 to 5 seconds). Two kind of cars have been defined, with maximum speeds of 50 and 80 km/h.

Table I summarizes the main simulation parameters. *Beacon interval* refers to the time between consecutive beacons. The information acquired is considered valid during the *beacon hold time*. Each run consists of one broadcasting task that is started by a random source chosen from a subset of vehicles that meet some requirements. Namely, the vehicle must be active when the steady state of the network is reached, and it must have at least 30 seconds remaining before reaching its destination (i.e., before leaving the simulation). The broadcasted message contains 500 bytes of payload and has a lifetime of 120 seconds, afterwards it is discarded. Results show the average value of 20 independent runs, along with the 95% confidence interval.

We focus on the following metrics:

- Reliability. Defined as the ratio between the number of vehicles which receive the broadcast message and the total number of them that could have received it: $Rel = N_{recv}/N_{total}, Rel \in [0, 1]$. Note that, probably, not every simulated node can receive the broadcasted message because some cars may remain partitioned from the source. In order to overcome this issue, we measure N_{total} on each simulation as follows. We have implemented and simulated with ideal MAC and PHY layers a variant of hyper-flooding [23]. The number of covered

---

[1] http://www.isi.edu/nsnam/ns/

[2] http://sumo.sourceforge.net/

nodes N_{recv} obtained on such simulations, becomes the upper bound N_{total} for the remaining protocols.

- Number of message transmissions per involved vehicle. This measures the efficiency of the protocol. Given the same reliability, a protocol is said to be more efficient than another if it needs fewer transmissions to complete the broadcasting task. The number of involved vehicles N_{total} has been computed as explained before.
- Control overhead per vehicle. Since the protocols are localized, the overhead comes from the periodic exchange of beacon messages. Our DV-CAST implementation also adds information as an extra header within data packets. The total number of bytes devoted to protocol information per simulated vehicle, during every run, has been measured.
- Delivery latency. Measured as the time, in seconds, since the data source issues the message until it arrives at every receiver. For this metric, we focus on one specific run.

In the following section we describe the low-level models employed (MAC, PHY and wireless signal propagation) and analyze their impact onto the protocols performance.

V. CHANNEL MODEL IMPACT

Simulation results may vary significantly depending on the assumptions that are made with respect to the communication links [25]. In order to assure that our solutions are robust and could perform well in real environments, we study the impact that these assumptions may have onto the protocols behavior, ranging from ideal conditions to quite realistic simulations.

A. UDG model

In the *unit disk graph* (UDG) model, two nodes u, v are neighbors and can therefore directly communicate iff $distance(u, v) \leq r$, where r is the radius of the communication range [14]. To simulate this, cars employ ideal MAC and PHY layers. So, collisions and interference from other transmissions are not taken into account. The wireless signal propagation model employs the free space Friis equation for communications at near distances. It switches to the *two-ray-ground* model for longer distances, capturing the higher attenuation of the ground reflection effect. In any case, the transmission power is configured to provide a fixed communication range of 250 meters, so that the successful decoding of a signal is deterministically determined. Therefore, this configuration is equivalent to the UDG model with a transmission radius $r = 250$, providing an ideal communication capability up to that distance.

B. IEEE 802.11p model with deterministic signal propagation (802.11p – TRG)

In this case, MAC and PHY layers correspond to the oncoming 802.11 standard amended for vehicular networks. The implementation is described in [24], and has been updated with the latest available patches[3]. Collisions and SINR levels are accounted to determine if a given frame can be correctly decoded or not. However, the deterministic *two-ray-ground* propagation model is still employed (fixed range of 250 meters).

C. IEEE 802.11p model with non-deterministic signal propagation (802.11p – NAK)

Deterministic propagation models are not able to capture some features of wireless signal propagation like multi-path fading [25][26]. The Nakagami distribution is often used to model the amplitude of a signal that reaches a receiver by multiple paths. Depending on the parameter μ (shape), fading from Rician to pre-Rayleigh (Rayleigh included) can be modeled by this distribution. Given that the amplitude is Nakagami-distributed with parameters (μ, ω), the signal power obeys a Gamma distribution with parameters $\left(\mu, \frac{\omega}{\mu}\right)$, where ω is an estimate of the average received power.

The former distribution has been employed to model fading in both highway [27] and suburban scenarios [28]. For the highway, the model is adjusted by means of a set of experiments carried out on highway 101 in the Bay Area. Estimate ω is obtained from a logarithmic path loss model, as implemented in [24]. In the suburban case we have implemented the Nakagami fading model and the dual-slope log-normal path loss model described in [28]. In this paper the authors conduct two set of experiments in a suburban area of Pittsburgh. The path loss and fading models have been configured with parameters adjusted to the second experiment described in such article (it has more data samples).

The transmission power at each transceiver has been configured as if this model provided a fixed communication range of 250 meters. However, this configuration provides a probabilistic communication range.

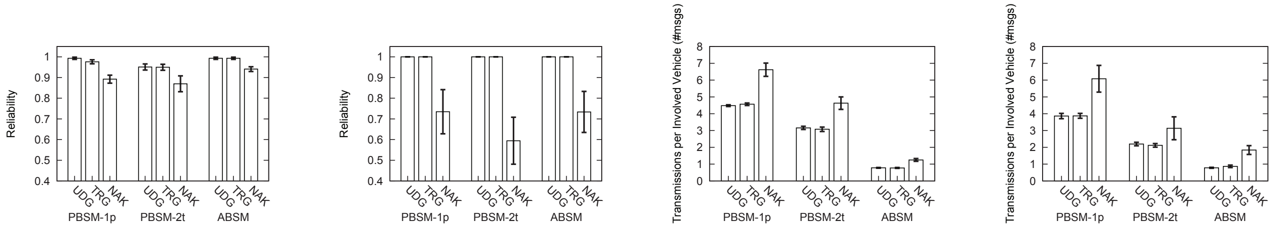
D. Results

We have simulated all the previously discussed models and some results for the most disconnected scenario (75 seconds between injected vehicles per route) are shown in Figure 3. We focus on this scenario because reliability results improve in general as the interval between injected vehicles decreases, since it means a more connected network. Therefore, all protocols, for each model, tend to achieve 100% reliability.

For this set of experiments, we fix $W = 0.25$ and $to\_ack = 1.6$ seconds. We can check that the MAC, PHY and channel models play a role in determining the final simulation results for every approach. In general, protocols performance is not very much affected when switching from an ideal UDG model to a deterministic 802.11p setup. Although in the latter case cumulative interference is accounted for determining whether a given frame can be successfully decoded, both configurations employ a fixed transmission radius of 250 meters. However, the more realistic non-deterministic channel models have a greater impact onto the results because of their random nature. Receivers inside the communication range might not decode the message, while others outside could do it with a non-negligible likelihood.

---

[3] http://dsn.tm.uni-karlsruhe.de/english/Overhaul_NS-2.php



(a) Reliability,Highway

(b) Reliability,Suburban

(c) Transmissions,Highway

(d) Transmissions,Suburban

Fig. 3. Reliability and number of data transmissions under different channel models with 75 seconds of traffic injection interval (95% confidence interval).

For all the evaluated protocols, the reliability in the highway scenario (Figure 3(a)) worsens as the model gets more realistic. This is especially true for the 802.11p – NAK model, which makes this metric decrease with respect to the UDG configuration by 10.1%, 8.1% and 5.2% for the PBSM-1p, PBSM-2t and ABSM protocols respectively. In an suburban environment differences across models are greater, as shown in Figure 3(b). While our protocols can achieve 100% reliability for UDG and 802.11p – TRG, the use of the Nakagami model can decrease the overall reliability by 36.4% (in the case of PBSM-2t). This lower performance with respect to the highway scenario is expected due to the higher fading experienced in urban deployments.

Given that the communication range is influenced by the low-level models, these also impact onto the number of data messages forwarded by the DS-NES based schemes. In particular, the probabilistic range provided by 802.11p – NAK makes every vehicle be aware of more neighbors (beacons might be decoded outside the “theoretical” radio range) and therefore the number of transmissions needed to cover the neighborhood increases (Figures 3(c) and 3(d)). In the highway, the number of transmissions per involved vehicle increases up to a factor of 1.4, 1.5 and 1.6 for PBSM-1p, PBSM-2t and ABSM when switching from UDG to 802.11p – NAK. In the densest simulated network (5 seconds of injection interval), this factor can reach the values 2.2, 2.0 and 2.8, respectively. For the suburban case, the realistic model makes the number of needed transmissions increase 1.6 (PBSM-1p), 1.4 (PBSM-2t) and 2.4 (ABSM) times in the scenario of Figure 3(d).

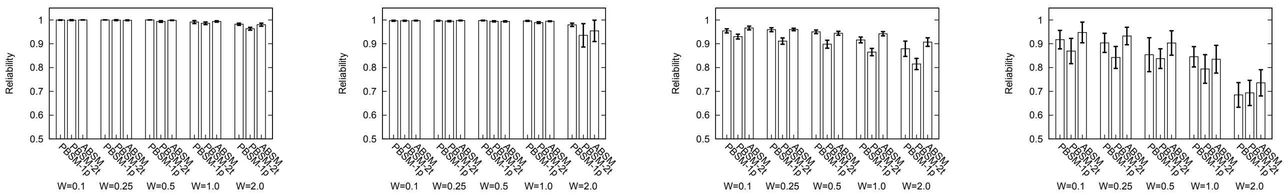
In all cases, ABSM achieves better or same reliability with much fewer data transmissions overhead. For the remainder of this paper, we focus on the most realistic 802.11p – NAK models in both highway and suburban environments.

VI. PARAMETERS SENSITIVITY ANALYSIS

As they have been defined, performance of our CDS-based protocols depends on a parameter to_ev (proportional to base time W) that represents the wait time before retransmitting a given message (see Eq. 1). In addition, ABSM incorporates an additional timeout to_ack which is employed to wait for acknowledgments of a forwarded message. They also rely on two more parameters, namely the beacon interval and holding time. However, the latter are related to to_ev and to_ack, which are the relevant parameters to study since they are exclusive of the evaluated broadcasting protocols. In this section we investigate how parameters W and to_ack influence the behavior of the protocols, fixing the beacon interval and holding time as shown in Table I.

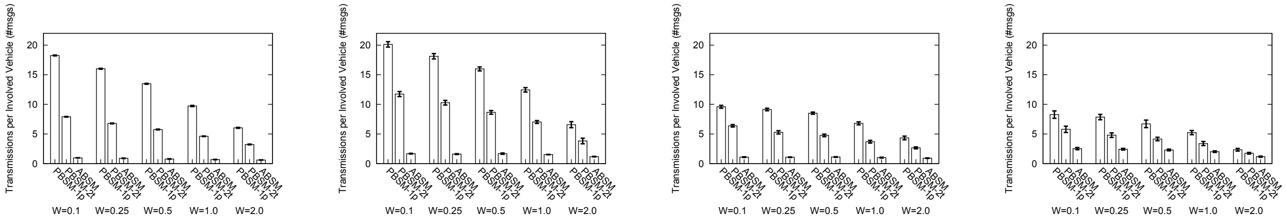
Figure 4 shows the impact of parameter W onto each protocol's reliability, where $to\_ack = 1.6$ seconds in the case of ABSM. We focus on a moderately dense network (30 seconds of injection interval) and a moderately sparse one (60 seconds of injection interval). Regardless the protocol under consideration, disconnected networks can benefit from low W values. This makes sense because some vehicles might remain as neighbors during a very short period of time. If the evaluation time is too high, the link between those vehicles might not exist any more and the forwarding opportunity would be lost. For denser networks, each protocol tends to converge at 100% reliability for every evaluated value of W. On the other hand, low W values generally provoke more redundant transmissions, especially in congested roads, as shown in Figure 5. As the waiting time before retransmission gets higher, the number of needed forwardings decreases. This phenomenon is explained by the use of NES, since the neighborhood might receive the message from other retransmissions. However, high W values augment the delivery latency of the broadcasting task, especially in sparse networks. This parameter is not relevant with respect to the protocol overhead. Taking these results into account, we recommend low W values for disconnected networks, and slightly higher values for dense ones. In any case, ABSM behaves very well in all the studied cases when compared to the other approaches.

We now fix $W = 0.25$ seconds and study the influence of to_ack onto ABSM performance. Delivery delay and protocol overhead are not very much affected by this parameter. With respect to protocol reliability, the evaluated figures for this parameter make such metric decrease at most 1.5% in the highway and 7% in the suburban section, when comparing $to\_ack = 0.6$ with $to\_ack = 2.1$. Intermediate values provide intermediate reliability. However, it makes a great difference with respect to the protocol efficiency (see Figure 6). to_ack controls how aggressive the acknowledgment strategy of ABSM is. Given that acknowledgments are borne in periodic beacons, this parameter must be at least slightly higher than the beacon interval. It has no sense retransmitting a message before the acknowledgment has been given opportunity of being issued. By making to_ack slightly higher than the beacon interval (0.6 seconds in our simulations), the protocol behaves aggressively: As soon as the first next beacon does not acknowledge the reception of the message, a retrans-



(a) Interval=30 sec,Highway (b) Interval=30 sec,Suburban (c) Interval=60 sec,Highway (d) Interval=60 sec,Suburban

Fig. 4. Effect of parameter W onto reliability for different traffic injection intervals and scenarios (95% confidence interval).



(a) Interval=30 sec,Highway (b) Interval=30 sec,Suburban (c) Interval=60 sec,Highway (d) Interval=60 sec,Suburban

Fig. 5. Effect of parameter W onto the number of data transmissions for different traffic injection intervals and scenarios (95% confidence interval).

mission takes place. Keep in mind that the message could be successfully received but the acknowledgment might have been lost due to a collision, jeopardizing the efficiency of the approach without improving the reliability of the broadcasting task. Thus, as we employ higher values of to_ack, ABSM is more efficient. As the simulation results show, increasing to_ack farther than the beacon holding time is also pointless. Therefore, the right choice for this parameter is always a value slightly higher than the beacon holding time.

In the following sections we stick with parameters $W = 0.25$ seconds and $to\_ack = 1.6$ seconds.

VII. PERFORMANCE COMPARISON

Let us focus on the comparative performance of ABSM with respect to the other PBSM variants and DV-CAST.

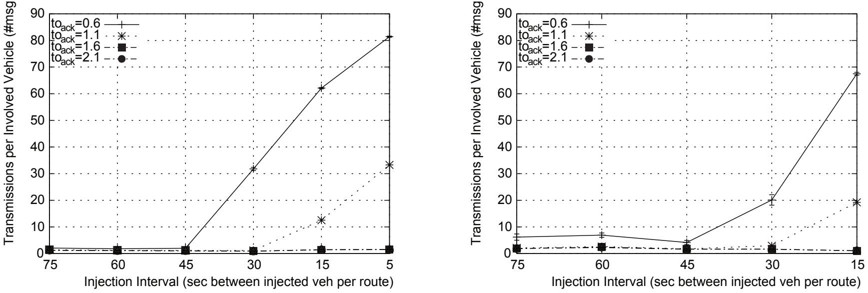
PBSM-1p, PBSM-2t and ABSM provide high reliability for broadcasting in highways (see Table II). This is not surprising because they are based upon the DS-NES forwarding framework, which is meant to cover the whole network. Among them, ABSM achieves the best results. The lowest reliability offered by this scheme is the 94.1% of the vehicles that could have received the message. Compared to PBSM variants, it achieves up to a 7% enhancement. On the other hand, DV-CAST offers a very poor reliability for sparse networks, while it only covers around the 75-85% of vehicles when the highest traffic rates are simulated. The reason is that the protocol does not foresee common vehicular movements such as passing maneuvers. For example, let us look again to Figure 1. Assume that vehicle f initiated the broadcasting and the message has been propagated backwards up to a (after applying the broadcast suppression technique). All vehicles are in idle state except a, which has the forwarding responsibility at that moment. Then, a speeds up and overtakes the remaining vehicles, forwarding the message to them and going to idle state. According to DV-CAST [6], the receivers discard the message as duplicated and the message custody is lost. No one will forward the message again even when new vehicles g, h emerge. This problem is derived from the different states in which DV-CAST operates depending on the traffic regime which is sensed by a vehicle, since it is hard to foresee every possible combination of movements in vehicular setups. Our approach does not suffer from this problem.

TABLE II
RELIABILITY RESULTS (%) ALONG WITH THE 95% CONFIDENCE INTERVALS IN THE HIGHWAY SCENARIO FOR DIFFERENT INJECTION INTERVALS (SEC BETWEEN INJECTED VEH PER ROUTE).

Interval	PBSM-1p	PBSM-2t	ABSM	DV-CAST
75	89.2 ± 1.9	87.0 ± 3.8	94.1 ± 1.1	16.7 ± 3.1
60	95.9 ± 0.8	91.1 ± 1.3	96.0 ± 0.5	20.8 ± 3.8
45	94.6 ± 0.7	92.5 ± 0.8	95.3 ± 0.6	20.2 ± 4.6
30	100 ± 0	99.9 ± 0.1	99.9 ± 0.1	63.4 ± 8.1
15	100 ± 0	100 ± 0	100 ± 0	72.6 ± 0.1
5	100 ± 0	100 ± 0	100 ± 0	85.7 ± 0.3

For the suburban scenario, Table III shows again that, in general, DS-NES approaches are reliable. Overall, ABSM outperforms the rest. In the scenario with lowest density ($75sec$ between injected vehicle) the reliability of our algorithms drops because the low density of vehicles provokes few contacts between them. When the protocol decides to forward the message, the receiver might not correctly receive it and the propagation opportunity is lost. This is related to the implicit randomness of the received signal power in the log-normal path loss and Nakagami fading models.

We focus now on the number of forwardings for each protocol, shown in Figure 7. Given the low reliability of DV-CAST, the number of broadcast messages issued by the protocol is also low. Interestingly, ABSM obtained the best reliability at the expense of almost as few transmissions as DV-CAST provokes. Furthermore, the number of broadcast messages issued by ABSM is almost constant with respect to the simulated traffic flow rate. This indicates the suitability of ABSM as a scalable solution for broadcasting in highways and



(a) ABSM,Highway

(b) ABSM,Suburban

Fig. 6. Effect of parameter to_ack onto the number of data transmissions for different traffic injection intervals (95% confidence interval).

TABLE III
RELIABILITY RESULTS (%) ALONG WITH THE 95% CONFIDENCE INTERVALS IN THE SUBURBAN SCENARIO FOR DIFFERENT INJECTION INTERVALS (SEC BETWEEN INJECTED VEH PER ROUTE).

Interval	PBSM-1p	PBSM-2t	ABSM
75	73.5 ± 10.7	59.4 ± 11.3	73.4 ± 9.9
60	90.4 ± 4.0	84.3 ± 4.6	93.3 ± 3.7
45	89.0 ± 10.1	92.3 ± 7.3	89.3 ± 10.0
30	99.7 ± 0.2	99.5 ± 0.3	99.8 ± 0.2
15	100 ± 0	100 ± 0	100 ± 0

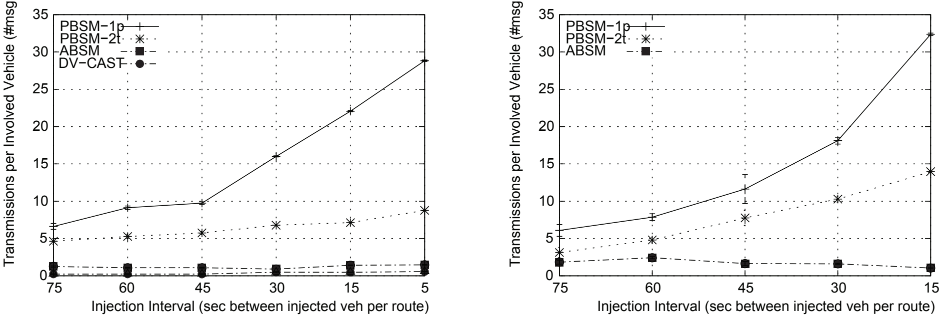
urban roads. It takes advantage of the piggybacked acknowledgments to reduce the protocol redundancy. When a node contacts a new neighbor for the first time, new forwardings are avoided if the latter already received the message. On the other hand, PBSM-1p and PBSM-2t need much more transmissions to obtain high reliability, especially when the network traffic is dense (more neighbors are discovered during the simulation). Since PBSM-2t works with 2-hop topology information at its disposal, it introduces more nodes into list R than PBSM-1p. This means more knowledge of receiver nodes, and therefore fewer transmissions. Interestingly, both PBSM-1p and PBSM-2t need many more than one retransmission per involved vehicle in order to achieve high reliability. However, in most scenarios ABSM needs around one forwarding per vehicle (sometimes less than one transmission). This is, it can achieve high reliability in disconnected networks without requiring every receiving node to retransmit the message.

So far, simulation results show that ABSM can achieve the best reliability results with the lowest number of transmissions. More important, the redundancy trend of the protocol remains almost constant as density of the network changes. Hence, ABSM scales with respect to this parameter. We also investigated the control overhead introduced in periodic beacon messages by each protocol. Figure 8 draws the values of this metric for different injection intervals. In general, PBSM-1p generates the lowest overhead, since it only adds the sender's position to the beacon. ABSM's overhead is slightly higher, because it also needs to include an identifier for each received broadcast message. Given the huge reduction in data message transmissions (Figure 7), ABSM is still the most efficient approach of all the evaluated ones. Our implementation of DV-CAST is heavier because direction information is added to the beacons. Additionally, it also includes control information inside data messages. Finally, the highest overhead is provoked by PBSM-2t, since it lists the whole 1-hop neighborhood within every beacon exchange.

We have also investigated the delivery latency that is experienced by each protocol. In order to check that ABSM does not penalize the message propagation delay, we focus on one run of a dense highway scenario (traffic injection interval is set to 15 seconds between vehicles per route) and measure the time since a message is generated until it is successfully decoded by every receiver. The results of this experiment are shown in Figure 9. We have ordered in time the reception of the message by each vehicle, have formed groups of 40 vehicles, and the average delay of each group is shown along the y-axis. It can be seen that ABSM and PBSM-1p deliver the message faster than the other approaches. PBSM-2t causes higher delivery latencies due to its increased overhead in periodic beacons, which means more contention and therefore longer times to access the medium. With respect to DV-CAST, it incurs larger delays under common retransmission parameter W. In the connected part of the network, the weighted p-persistence broadcast suppression technique is applied. Hence, the waiting time before retransmission is constant (either W or $W + \delta$), contrary to our adaptive approach in which this value depends on the local density of the network. Besides, when there are disconnected groups of vehicles that eventually merge, there is an increased latency for every protocol. However, this is higher in DV-CAST because only a subset of vehicles that own the message custody are the ones that can forward it. Unfortunately, other vehicles might have delivered it quicker to the approaching nodes, as it is the case in A/PBSM. Figure 9 also shows that DV-CAST reaches fewer vehicles than our solutions, since it can not reach group of vehicles 161–200 and 201–240.

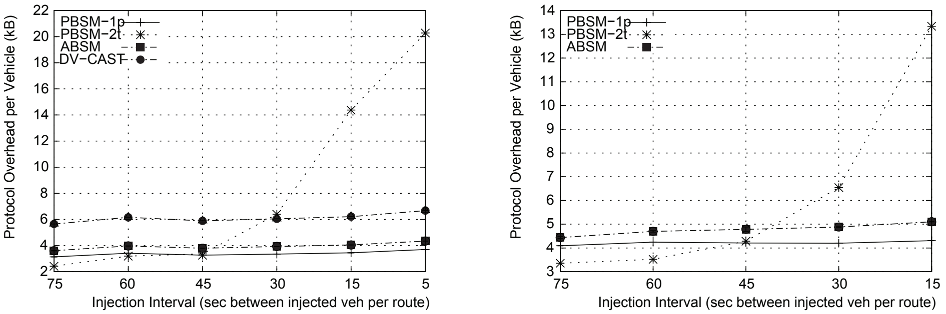
VIII. BENCHMARKING

In this section we are interested in checking how far ABSM is of finding the optimum number of transmissions to complete a broadcasting task. We model the dynamic vehicular network as an evolving graph $\mathcal{G} = (V_\mathcal{G}, E_\mathcal{G})$, where each node $v \in V_\mathcal{G}$ and each link $e \in E_\mathcal{G}$ has a presence schedule defined for it. Presence schedules indicate the intervals at which a given



(a) Highway (b) Suburban

Fig. 7. Number of data transmissions per involved vehicle (95% confidence interval).



(a) Highway (b) Suburban

Fig. 8. Control overhead (kBytes) per vehicle (95% confidence interval).

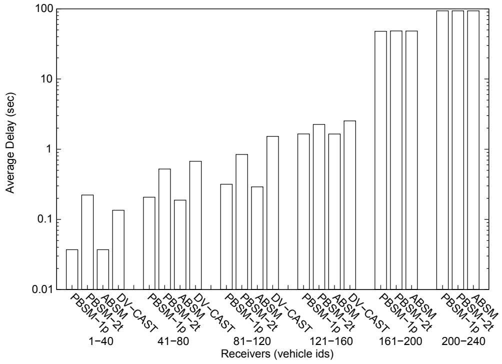


Fig. 9. Delivery latency (sec) for every receiver in the highway scenario with 15 seconds of traffic injection interval. Logarithmic scale along the y-axis. Vehicles are grouped from the first ones that receive the message to the last ones, and the average delay of each group is shown.

node or link exists, and therefore incorporate the time domain into the notion of graph. One can think of an evolving graph as a set of graphs indexed in time. Every vehicle is a vertex in $V_{\mathcal{G}}$, and edges $E_{\mathcal{G}}$ are determined by applying the UDG model at every time instant.

Finding the minimum number of transmissions to complete a broadcasting task in a given graph G is an NP problem [14]. Obviously, it is also NP in an evolving graph $\mathcal{G}$ because this problem reduces to the former by assuming a single time instant. Therefore, there exists no efficient solution that can be applied here. We have implemented a centralized algorithm that finds the optimum number of transmissions in $\mathcal{G}$ given a data source s. It generates an exploration tree in which every level represents a time instant, and every new branch explores a combination of nodes that can forward the message. A solution is found when all nodes have received the message, and the one with the fewer number of transmitters is returned as the optimum. To save execution time, we employ the branch-and-bound technique to reject those branches that do not lead to an optimal solution. Initially only the source holds the message, while the remaining simulated vehicles are in need of it. At each step, the algorithm operates with the graph G_t corresponding to the current time instant. It generates all the possible combinations of forwarders among those vehicles that hold the message. The neighbors in G_t of these forwarders are moved from the set of pending nodes in need of the message to the set of vehicles that received it. The time instant is advanced and the same operations performed, until the set of pending nodes becomes empty (a solution has been found). The remaining cases are still evaluated, since we need to explore the whole tree to return the solution that provokes the lowest number of transmissions. We bound those branches in which the number of transmissions is equal or greater than the current optimum, and those in which a transmission of a node does not mean including new vehicles into the set of nodes that hold the message.

Despite the applied optimizations, the algorithm takes too much execution time. Therefore, we have focused on a simple

TABLE IV
SIMULATION PARAMETERS FOR BENCHMARKING.

Simulation Time	45 sec
Broadcasting Start	27.5 sec
Traffic Rate	[1/18, 1/14, 1/10, 1/6] veh/sec/route
Maximum Speed	80 ± 20 km/h
Runs	20 (95% confidence interval)

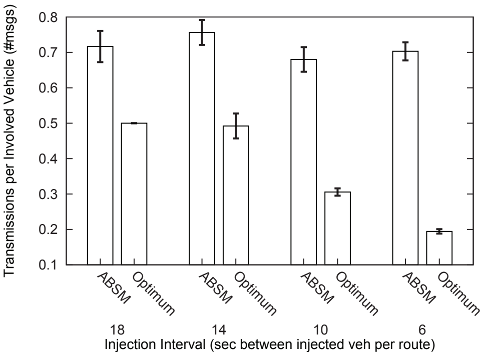


Fig. 10. Comparison of ABSM against the optimum number of transmissions for different injection intervals (95% confidence interval).

highway (1 km long, one lane per direction) with few vehicles. We also simulate ABSM on such scenario assuming a UDG model, and compare the obtained number of forwardings with the optimum with respect to this metric. Simulation parameters for this experiment are as in previous section except those shown in Table IV. The data source has been fixed to be the first simulated vehicle. Figure 10 shows that, as expected, ABSM is not able to cover the whole network with as few forwardings as the optimum solution. Clearly, ABSM is a distributed protocol that only relies on local position information, meanwhile to discover the optimum we need complete topology information for every time instant (including knowing how the network evolves in the future). However, we can see that there are little differences between ABSM and the optimum in sparse networks. The rate between data transmissions and the number of simulated vehicles remains almost constant in ABSM as more vehicles are injected into the network. It issues more retransmissions because more nodes in need of the message are discovered. The optimum algorithm is able to anticipate the future and forward the message only when the highest number of neighbors would receive it, therefore minimizing the number of transmissions. When the density increases, the rate of needed forwarders decreases because more vehicles receive the message with each transmission.

IX. CONCLUSIONS AND FUTURE WORK

We have presented ABSM, a localized broadcast protocol for vehicular ad-hoc networks. It is built upon the DS-NES framework, and implicitly uses the store-carry-forward paradigm typical of delay-tolerant networks. The algorithm employs the position information of the 1-hop neighborhood, as well as acknowledgments of the latest received broadcast messages, to improve the protocol reliability and efficiency.

By means of a thorough simulation-based study, ABSM has been evaluated in highway and urban scenarios. Different mobility degrees and network densities have been taken into account. In addition, we considered different models for wireless technologies, ranging from ideal conditions to quite realistic 802.11p simulations. We further analyzed the protocol by performing a sensitivity analysis on the parameters it depends on, and studying ABSM's scalability as the number of data sources in the network increases. The proposed algorithm has been shown to outperform competing solution specifically designed for the vehicular environment. ABSM has turned out to be a very robust and reliable protocol, that extremely reduces the number of transmissions needed to complete a broadcasting task. Despite the algorithm is delay-tolerant by nature, it does favor low delivery latencies.

The results provided within this paper are very promising and encouraging. We plan to continue working on ABSM in the vehicular context. We will address the degree of compatibility of the protocol with developing standards like DSRC. The protocol will also be analyzed when infrastructure nodes also take part in data messages dissemination. On the other hand, we are currently investigating how to further reduce the protocol overhead when there are multiple simultaneous broadcasting tasks, by means of probabilistic data structures to limit the size of the acknowledgment list in beacon messages. In addition, we will also investigate how to adapt the retransmission timeout given a delay constraint from the application, in order to make the protocol suitable to delay-critical safety applications.

ACKNOWLEDGMENT

This work was partially supported by NSERC Discovery grant, NSERC Strategic Grant STPSC356913-2007B and the CENIT MARTA project with funding from CDTI supported by the Spanish Science and Innovation Ministry.

REFERENCES

[1] H. Hartenstein and K. Laberteaux, A tutorial survey on vehicular ad hoc networks, *IEEE Communications Magazine*, Vol. 46, Issue 6, pp. 164–171, 2008.

[2] M. Sichitiu and M. Kihl, Inter-vehicle communication systems: a survey, *IEEE Communications Surveys & Tutorials*, Vol. 10, Issue 2, pp. 88–105, 2008.

[3] U. Lee, J. Lee, J. Park, E. Amir and M. Gerla, FleaNet: A Virtual Market Place on Vehicular Networks, in *Proc. of Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services*, pp. 1–8, Jul. 2006.

[4] M. Sun, W. Feng, T. Lai, K. Yamada, H. Okada and K. Fujimura. Gps-based message broadcast for adaptive inter-vehicle communications, in *Proc. of 52nd IEEE Vehicular Technology Conference (VTC)*, Vol. 6, pp. 2685–2692, 2000.

[5] S. Biswas, R. Tatchikou and F. Dion, Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety, *IEEE Communications Magazine*, Vol. 44, Issue 1, pp. 74–82, Jan. 2006.

[6] O. Tonguz, N. Wisitpongphan, F. Bai, P. Mudalige and V. Sadekar, Broadcasting in VANET, in *Proc. of Mobile Networking for Vehicular Environments*, pp. 7–12, Anchorage, AK, USA, May 2007.

[7] G. Korkmaz, E. Ekici, F. Ozguner and U. Ozguner, Urban multi-hop broadcast protocol for inter-vehicle communication systems, in *Proc. of the 1st ACM International Workshop on Vehicular Ad Hoc Networks (VANET)*, pp. 76–85, Oct. 2004.

[8] G. Korkmaz, E. Ekici and F. Ozguner, An efficient fully ad-hoc multi-hop broadcast protocol for inter-vehicular communication systems, in *Proc. of IEEE International Conference on Communications (ICC)*, Jun. 2006.

[9] F. Ros, P. Ruiz and I. Stojmenovic, Reliable and Efficient Broadcasting in Vehicular Ad Hoc Networks, in *Proc. of 69th IEEE Vehicular Technology Conference (VTC)*, Barcelona, Spain, Apr. 2009.

[10] I. Stojmenovic, M. Seddigh and J. Zunic, Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 13, No. 1, pp. 14–25, Jan. 2002.

[11] I. Stojmenovic, Comments and Corrections to “Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks”, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 15, No. 11, pp. 1054–1055, Nov. 2004.

[12] IEEE P802.11p/D0.21, Draft Amendment to Standard for Information Technology Telecommunications and Information Exchange Between Systems LAN/MAN Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Wireless Access in Vehicular Environments (WAVE), Jan. 2006.

[13] I. Stojmenovic and J. Wu, Broadcasting and activity scheduling in ad hoc networks, in *Mobile Ad Hoc Networking* (S. Basagni, M. Conti, S. Giordano and I. Stojmenovic, eds.), IEEE Press, pp. 205–229, 2004.

[14] B. Clark, C. Colbourn and D. Johnson, Unit disk graphs, *Discrete Mathematics*, Vol. 86, Issue 1-3, pp. 165–177, 1991.

[15] J. Wu and H. Li, A dominating set based routing scheme in ad hoc wireless networks, *Telecommunication Systems*, Vol. 18, No. 1-2, pp. 13–36, 2001.

[16] W. Peng and X. Lu, On the reduction of broadcast redundancy in mobile ad hoc networks, in *Proc. of ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Boston, USA, 2000.

[17] A. Khan, I. Stojmenovic and N. Zaguia, Parameterless broadcasting in static to highly mobile wireless ad hoc, sensor and actuator networks, in *Proc. of 22nd IEEE International Conference on Advanced Information Networking and Applications (AINA)*, Ginowan, Okinawa, Japan, Mar. 2008.

[18] M. Li, W. Lou and K. Zeng, OppCast: Opportunistic Broadcast of Warning Messages in VANETs with Unreliable Links, in *Proc. of the 6th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, Oct. 2009.

[19] M. Nekovee, Epidemic algorithms for reliable and efficient information dissemination in vehicular ad hoc networks, *IET Intelligent Transportation Systems*, Vol. 3, Issue 2, pp. 104–110, 2009.

[20] N. Wisitpongphan, O. Tonguz, J. Parikh, P. Mudalige, F. Bai and V. Sadekar, Broadcast storm mitigation techniques in vehicular ad hoc wireless networks, *IEEE Wireless Communications Magazine*, pp. 84–94, Dec. 2007.

[21] Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems - 5 GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Oct. 2002.

[22] C. Lochert, M. Mauve, H. Fussler and H. Hartenstein, Geographic routing in city scenarios, in *ACM SIGMOBILE Mobile Computing and Communications Review*, Vol. 9, No. 1, pp. 69–72, 2005.

[23] K. Viswanath and K. Obraczka, An adaptive approach to group communications in multi-hop ad hoc networks, in *Proc. of IEEE International Symposium on Computers and Communications (ISCC)*, pp. 559–566, Taormina, Italy, 2002.

[24] Q. Chen, F. Schmidt-Eisenlohr, D. Jiang, M. Torrent-Moreno, L. Delgrossi and H. Hartenstein, Overhaul of IEEE 802.11 modeling and simulation in NS-2, in *Proc. of the 10th ACM/IEEE International Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM)*, pp. 159–168, Chania, Greece, Oct. 2007.

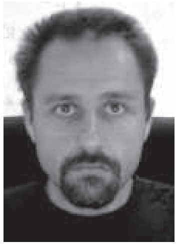
[25] M. Takai, J. Martin and R. Bagrodia, Effects of Wireless Physical Layer Modeling in Mobile Ad Hoc Networks, in *Proc. of ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Oct. 2001.

[26] M. Torrent-Moreno, D. Jiang and H. Hartenstein, Broadcast reception rates and effects of priority access in 802.11-based vehicular ad-hoc networks, in *Proc. of the 1st ACM International Workshop on Vehicular Ad Hoc Networks (VANET)*, pp. 10–18, Oct. 2004.

[27] V. Taliwal, D. Jiang, H. Mangold, C. Chen and R. Sengupta, Empirical determination of channel characteristics for DSRC vehicle-to-vehicle communication, in *Proc. of the 1st ACM International Workshop on Vehicular Ad Hoc Networks (VANET)*, pp. 88–88, Oct. 2004.

[28] L. Cheng, B. Henty, D. Stancil, F. Bai and P. Mudalige, Mobile Vehicle-to-Vehicle Narrow-Band Channel Measurement and Characterization of the 5.9 GHz Dedicated Short Range Communication (DSRC) Frequency Band, *IEEE Journal on Selected Areas in Communications*, Vol. 25, No. 8, pp. 1501–1516, Oct. 2007.

Francisco J. Ros is currently a Ph.D. candidate at the University of Murcia (Spain). He received a B.Sc. degree in Computer Science (2004), a Postgraduate diploma in New Technologies in Information and Communications Engineering (2007), and a Master’s degree in Advanced Information and Telematic Technologies (2009) from the same University. He is currently working as a researcher and part-time adjunct professor for the Department of Information and Communications Engineering (DIIC). Ros is a technical committee member of ADHOC-NOW (2009&2010), and serves as a reviewer in major IEEE journals and conferences. His main research interests include ad-hoc networking, network performance modeling, distributed algorithms, and new generation networks.



Pedro M. Ruiz received his B.Sc. (1999), M.Sc. (2001) and Ph.D. (2002) degrees in Computer Science from the University of Murcia, Spain. He works as *Associate Professor* in Telematics at the Department of Information and Communication Engineering (DIIC) at the University of Murcia (UMU). In 2003 he was awarded a *Ramón y Cajal* research position by the Spanish MEC. He has also held Post-doctoral research positions at ICSI in Berkeley, King’s College London and University of California at Santa Cruz. During these years he has acted as Principal Investigator in a number of research projects mainly funded by the European Union, Spanish government and private companies, and has published a large number of refereed papers in international journals and conferences. Dr. Ruiz received in 2007 an outstanding research trajectory recognition from the Spanish MEC. He is in the editorial board for the International Journal on Parallel, Emergent, and Distributed Systems, International Journal of Network Management and International Journal on Smart Home. He has organized serveral workshops on localized algorithms and protocols co-located with IEEE MASS, ACM MobiHoc and IEEE DCOSS. He has also served in a number of Organizing and Technical Committees of highly relevant conferences such as ACM MobiCom, IEEE MASS, ACM MobiHoc, IEEE SECON, etc. He also serves as a reviewer for major IEEE journals and conferences. His main research interests include vehicular networks (VANET), sensor networks, mobile and ad hoc wireless networks and distributed systems. He is a member of the ACM and IEEE Communications Society.

Ivan Stojmenovic received his Ph.D. degree in mathematics. He held regular and visiting positions in Serbia, Japan, USA, Canada, France, Mexico, Spain, UK (as Chair in Applied Computing at the University of Birmingham), Hong Kong, Brazil, Taiwan, and China, and is Full Professor at the University of Ottawa, Canada and Adjunct Professor at the University of Novi Sad, Serbia. He published over 250 different papers, and edited seven books on wireless, ad hoc, sensor and actuator networks and applied algorithms with Wiley. He is editor of over dozen journals, editor-in-chief of IEEE Transactions on Parallel and Distributed Systems (from January 2010), and founder and editor-in-chief of three journals (MVLSC, IJPEDS and AHSWN). Stojmenovic is one of $\approx$ 260 computer science researchers with h-index $\geq$40 and has $>$7000 citations. He received three best paper awards and the Fast Breaking Paper for October 2003, by Thomson ISI ESI. He is recipient of the Royal Society Research Merit Award, UK. He is elected to IEEE Fellow status (Communications Society, class 2008), and is IEEE CS Distinguished Visitor 2010-12. He received Excellence in Research Award of the University of Ottawa 2009. Stojmenovic chaired and/or organized $>$60 workshops and conferences, and served in $>$200 program committees. He was program co-chair at IEEE PIMRC 2008, IEEE AINA-07, IEEE MASS-04&07, EUC-05&08-10, AdHocNow08, IFIP WSAN08, WONS-05, MSN-05&06, ISPA-05&07, founded workshop series at IEEE MASS, ICDCS, DCOSS, WoWMoM, ACM Mobihoc, IEEE/ACM CPSCom, FCST, MSN, and is/was Workshop Chair at IEEE INFOCOM 2011, IEEE MASS-09, ACM Mobihoc-07&08.