Not All VANET Broadcasts Are the Same: Context-Aware Class Based Broadcast

Falko Dressler, *Fellow, IEEE*, Florian Klingler, *Member, IEEE*, Christoph Sommer, *Member, IEEE*, and Reuven Cohen, *Senior Member, IEEE*

***Abstract*— A major building block of Vehicular Ad Hoc Networks (VANETs) is broadcasting: the use of wireless communication for sharing information among vehicles, or between the vehicles and infrastructure. Dozens of broadcast protocols have been developed in recent years, including protocols for 1-hop broadcasting of vehicle status information (beaconing) and for geocasting-based applications. However, most of these protocols were designed for one application and cannot co-exist, nor can one broadcast solution meet the demands of all applications. These observations motivated our effort to develop a holistic network layer for VANETs. We identify the need for making VANET broadcast context-aware, and for supporting four different classes of broadcast protocols, each with its own properties. These classes are not only able to co-exist on the same network layer, but also to complement one another's functionality. Thus, large applications as well as more holistic Transport protocols can be designed by combining two or more broadcast classes. We discuss the specific characteristics of these classes and design candidate protocols for each class.**

***Index Terms*— Vehicular networking, broadcasting, information dissemination, protocol design.**

I. Introduction

AFTER a decade of vehicular networking research [1]–[3], and although many important problems on the way towards a holistic Network layer for Inter-Vehicle Communication (IVC) are still unsolved [4], we are close to seeing the first real-world applications that are based on the IEEE 802.11p/DSRC standard [5]. The U.S. Department of Transportation (US DOT) works towards making Dedicated Short Range Communication (DSRC) radios mandatory in new cars. All major car makers in Europe joined a self-commitment towards this objective. The basic day-one application protocols are currently being field tested both in the US and in Europe.

Broadcasting[1] is the main communication primitive in vehicular networking for two main reasons. First, almost all VANET applications need to share the same piece of information with many vehicles in some area. Secondly, also for multihop dissemination most approaches rely on multiple forwarders [6], making broadcast the natural form of communication. IP and IPv6 based solutions are currently investigated by the IETF [7] but mainly for unicast applications in the non-safety domain. Unicast has been shown to cause even more problems in VANETs due to the head-of-line blocking problem of IEEE 802.11 in very dynamic environments [8]. Current approaches to broadcast protocols follow the "one-fits-all" approach: they employ a single, often beaconing-based, broadcast protocol to support all envisioned VANET applications. However, when we study the properties of such applications, we soon see that they can be optimally supported *only using a specialized Network layer that employs several different broadcast protocols*. Indeed, in recent years we have witnessed many proposals for VANET broadcast protocols, each designed with a specific application in mind: platooning, intersection safety, cooperative awareness, traffic information, etc [9]–[13]. But all these protocols were proposed by different research groups, and were not intended to cooperate or even to co-exist on the same Network layer.

We carefully investigate the differences and commonalities of VANET broadcast protocols and identify that *not all VANET broadcasts are the same*. Moreover, we distinguish a set of four classes of broadcast protocols that we believe would suit all VANET applications, ranging from ultra-low latency safety to generic range-oriented geocasting solutions. The protocols in each class must be context-aware, namely, their basic properties depend on the application requirements. Thus, these protocols differ greatly in the number of nodes (vehicles) that are likely to invoke each broadcast protocol, in the coverage and reliability (retransmission strategy) of each protocol, in the priority of the messages created by each protocol, and so on.

In this paper, we propose a novel, integrated, context-aware, broadcast-based Network layer for supporting past and future VANET applications. We further propose four broadcast classes (called A to D in this paper) that match the requirements of all known applications. These classes co-exist on the same Network layer, and also make use of cross-protocol functionality. For example, a protocol from one class relies on the information provided by the protocol from another class, or an event is first broadcast by a (real-time) protocol from one class and then by a (non-real-time) protocol from another. Our findings clearly show a strong dependency between the various broadcast classes. The proposed broadcast classes represent the underlying basis for designing new applications and more holistic Transport protocols by combining

Manuscript received September 26, 2016; revised July 4, 2017; accepted September 28, 2017; approved by IEEE/ACM Transactions on Networking Editor Y. Yi. Date of publication October 30, 2017; date of current version February 14, 2018. *(Corresponding author: Falko Dressler.)*

F. Dressler, F. Klingler, and C. Sommer are with the Department of Computer Science, Paderborn University, 33098 Paderborn, Germany (e-mail: dressler@ccs-labs.org; klingler@ccs-labs.org; sommer@ccs-labs.org.

R. Cohen is with the Department of Computer Science, Technion, Haifa 3200003, Israel (e-mail: rcohen@cs.technion.ac.il).

Digital Object Identifier 10.1109/TNET.2017.2763185

[1]We use the term broadcast also for multicast or geocast protocols as their main principle is to broadcast on the wireless medium and the receiving nodes decide about further processing in a fully decentralized manner.

two or more classes. We believe that our key findings for each class of protocols build the basis for future protocol design.

Our key contributions can be summarized as follows:

- We develop a novel context-aware class-based broadcasting framework for the Network layer of VANETs, which consists of four different classes of broadcast protocols (Section III).
- We design candidate protocols for all classes, or select suitable protocols from the relevant literature (Section IV).
- We provide a detailed analysis of the performance of the various proposed protocols (Section V) as well as how they influence each other when being executed concurrently.

II. Fundamentals and Related Work

Recently, a variety of generalized protocol stacks have been proposed, supporting a full protocol stack for different vehicular networking applications [1]. As application requirements are not always the same, dedicated broadcast protocols have been investigated for use with a single, very specific application. These protocols almost always require a dedicated radio channel for operation, i.e., one that supports only one application.

A. Generalized Protocols

Many generalized protocol stacks focus on a single application domain, *cooperative awareness*, but make the underlying broadcast protocol also available for other applications [14]. For this, Cooperative Awareness Messages (CAMs), also known as Basic Safety Messages (BSMs), are broadcast at fixed intervals, usually in the range of 0.1-1 s, a procedure also known as *beaconing* [12], [15]. Since the communication in VANETs is not sufficiently reliable, approaches that increase reliability while minimizing the number of beacon retransmissions have been investigated [16]. The main challenge, which holds for all classes of broadcast based protocols, is that the frequency of possible broadcasts strongly depends on the available capacity of the wireless channel. Thus, the protocol must be able to prevent broadcast storms [17]. To this end, adaptive beaconing solutions that focus on congestion control have been developed [18], [19]. Later on, fairness as a primary objective was integrated with these adaptive solutions [20].

One of the first approaches, Adaptive Traffic Beacon (ATB) [19], addresses exactly these problems by focusing on two main questions: How frequently can the protocol send beacons; and how frequently should the protocol send beacons? For this, two different metrics have been introduced: the *channel quality* C and the *message utility* P, to calculate the beacon interval I with which to disseminate messages. ATB adjusts I such that it becomes minimal only for the highest message utility and the best channel quality; in all other cases, channel use is reduced drastically, allowing uninterrupted use of the channel by other applications.

Furthermore, transmit power control can be employed to increase spatial reuse [21]. It has been shown, however, that transmit power control might be counterproductive for safety-critical applications [22].

These ideas have been picked also by standardization bodies. ETSI ITS-G5 now defines a standardized beaconing protocol, which adapts – via Decentralized Congestion Control (DCC) Transmit Rate Control (TRC) [23] – the inter-beacon interval according to a state machine consisting of independent meta states referring to different beaconing intervals I_{min}, I_{def}, and I_{max}. TRC periodically measures the channel busy ratio b_t by using a complex sampling process and thus performs the necessary transitions in the state machine. Two parameters b_{up} and b_{down} are used to decide whether to increase or decrease the inter beacon interval. This leads the protocol to react to overloaded channels, but at the same time to use the lowest available beaconing interval whenever possible.

Pulsar [24] as an alternative uses 2-hop piggybacking of congestion control information in order to maximize the overall beacon rate of nodes – specifically used for vehicular safety applications. LIMERIC [25] represents a novel adaptive congestion control algorithm to adapt the beacon rate in order to provide fairness to all nodes. A further study [26] shows that a more aggressive approach is especially beneficial for abrupt topology changes as well as low latency applications.

Bloom filters [27] have been investigated for network applications [28]–[31] to address congestion on the wireless channel besides the already mentioned adaptive beaconing solutions. However, these approaches target very specific application scenarios and did not focus on generic neighborship management, which is very important for VANETs.

A first approach towards 2-hop neighbor information dissemination using Bloom filters has been presented very recently [32]. However, this approach uses the intersection of several Bloom filters, which increases the false positive error probability, thus causing loss of information. In comparison to traditional approaches not using Bloom filters, the presented work reveals that the beacon overhead could be decreased but no substantial Application layer performance gain can be achieved.

Geocasting goes one step beyond simple 1-hop broadcasting [33]. It combines broadcast with geographical knowledge, and fulfills many additional application requirements [34], [35]. More recent protocol designs combine geocasting with Delay Tolerant Networking (DTN) capabilities [36] and exploit estimated vehicle trajectories [37].

All of these approaches follow a one-fits-all concept, which is very limited in its suitability for all possible IVC applications. Therefore, substantial research towards application-specific broadcast protocols has been conducted.

B. Application-Specific Protocols

Application-specific protocols have been investigated in a whole spectrum of potential applications, of which we choose several examples to illustrate the requirements of the selected approaches. We start again with cooperative awareness. This application also relies on multi-hop broadcasting in order to counter the substantial radio signal shadowing that occurs in urban environments. A reliable broadcast protocol that touches on this problem is published in [38], and the use

TABLE I
THE PROPOSED BROADCAST CLASSES

Class	message priority	expected number of initiators	scope of the broadcast	expiration time	broadcast type	report merging
A	normal	only 1	1-hop transmission circle	less than a second	unreliable	no
B	very high	very few (1-4)	N-hop transmission distance	less than a second	semi-reliable	no
C	high	few or more (1-20)	a certain geographic area	several seconds	semi-reliable	yes
D	low	few to many (1-100)	a certain geographic area	several minutes	reliable	yes

of parked vehicles to increase reliability has been proposed in [39] and [40].

Another application is Cooperative Adaptive Cruise Control (CACC), or platooning. This application has possibly the tightest real-time requirements for maintaining cruise control. Broadcast based communication protocols have been designed for this application. These protocols require a dedicated radio channel for operation [9]–[11], [41].

Information of a less time-critical nature is exchanged in road traffic information systems, information downloading, and vehicular cloud applications. Broadcast protocols have been defined for all these application classes, with a primary focus on providing a certain degree of reliability [12], [13].

All the mentioned solutions cannot easily be combined with other protocols – there is a strong need for dedicated channels. This obviously limits the applicability of the developed protocols depending on the geographic region, as at most five available service channels have been dedicated for use in the vehicular networking context [42].

In this paper, we aim at combining both approaches, the generalized and the application-specific protocol design approach, by identifying the need for different classes of broadcast protocols. We developed an integrated Network layer based on four broadcast classes in a context-aware approach.

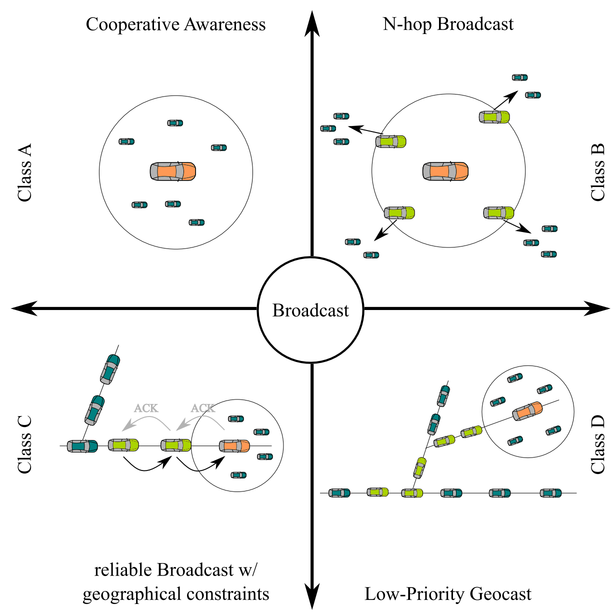


Fig. 1. Our vision of VANET broadcast classes: Class A for medium priority CAMs; Class B to forward highest priority events within N hops; Class C for high priority reliable broadcasting with geographical constraints; and Class D for low priority geocasting.

III. CLASS BASED BROADCASTS

We start our discussion of the Class Based Broadcast concept with a list of classification criteria for VANET broadcast protocols. This list will serve as the basis for our proposed broadcast classes. Next we present details about the four broadcast classes, and finally show the system design of our holistic Network layer.

A. Classification Criteria

We identified the following three main classification criteria and other, less important, criteria as indicated in Table I.

The first classification criterion is the *priority* of the event that triggers the broadcast. We distinguish between routine events, such as broadcasting a beacon message periodically in order to detect all 1-hop neighbors, and extraordinary events, such as announcing the detection of an object or an animal on the road. In addition, there are events that are not periodic, but also not very extraordinary, such as detecting a free parking lot. This criterion is important because it affects the priority of the broadcast messages.

The second classification criterion is the *expected number of vehicles* (nodes) that are likely to detect an event. For example, if a vehicle experiences a mechanical problem, it is the only node to be aware of this event. Congestion on the highway, however, is likely to be detected by all the vehicles driving in the reverse direction. When many nodes detect and report the same event, the channel might become heavily loaded, and many wireless collisions are likely to take place. This must be handled by the broadcast protocol.

The third classification criterion is the *scope* or target of the broadcast message. The scope of the broadcast is crucial for deciding how messages will be propagated from one vehicle to another and how this propagation will be stopped. One option is for the broadcast to target all the nodes within a given geographic radius (e.g., $1\,\mathrm{km}$) from the broadcast originator. This is likely to be the case when a vehicle invokes the broadcast to report the detection of a free parking lot in the center of town, or to report a traffic jam in a relatively distant area (e.g., $5\,\mathrm{km}$ from the location of the reporting vehicle).

B. Broadcast Classes

We use the above criteria to define four VANET broadcast classes as depicted in Figure 1. The main distinctions between these classes are outlined in Table I. In Section IV we will define specific protocols for the classes we propose.

1) Class A: This class consists of beaconing protocols, which broadcast periodic CAM or HELLO messages to 1-hop neighbors. In this class there is only one initiator for each broadcast. Current standardization efforts in the scope of

IEEE 1609 as well as ETSI ITS-G5 focus on this class of protocols. The periodic nature of this class allows each node to maintain an up-to-date list of its neighbors. Initially, this type of protocol has been considered for cooperative awareness applications only. We show in this paper that, in fact, it also allows maintaining information about 2-hop neighbors, which can be useful for the operation of the other broadcast classes. Class A protocols serve as basis for all other classes of protocols by providing this 2-hop neighbor information.

2) Class B: This class consists of protocols that broadcast information about an emergency event that is likely to be detected only by a few vehicles. Examples include information about an animal on the road, a broken-down vehicle, or a sudden stop. Broadcast protocols in this class should cover all the vehicles that surround the detecting vehicle and are not too far from it, i.e., following vehicles on a freeway or vehicles approaching the same intersection. The Decentralized Environmental Notification Message (DENM) concept is very close to this class [35], but there is one important difference as follows. DENM provides geocasting capabilities, which delay the propagation of the messages due to the need to perform time-consuming computations at the Application layer forwarding nodes. Since Class B messages are of high priority, we believe that it would be better not to identify the covered area using GPS coordinates.

Our alternative approach is that the covered vehicles be within N-hop radio transmission distance from the originator. The value of N is typically 1 or 2, depending on the event, and is determined by the application that detects and announces the event. Protocols in this class should not try to merge reports that are originated by different nodes for the same event, first because the number of nodes that detect a Class B event is small, and second because merging different reports requires that the content of different messages be queued and compared in the Application layer. This content comparison substantially delays the speed of broadcast. Different messages about the same event that are originated by the same source are identified by their Network layer headers and are pruned from the network in order to reduce the communication cost. The unique feature of Class B protocols is that they do not require geographical positions; the dissemination range is N hops.

3) Class C: This class consists of protocols that broadcast information about ongoing important-but-not-very-urgent events that are likely to be detected by many nodes. The reported event in Class C is also relevant to nodes that are much farther away from the detecting node, which is usually not the case for an event reported by Class B. Moreover, the exact geographic area depends on the type of the event. Thus, the reporting node needs to determine the geographic coordinates of the area to which the reported event is relevant.

Protocols in this class must prevent a broadcast storm [17]. This is typically done in two ways: First, a node that identifies an event initiates a new broadcast process with probability $p \leq 1$. Second, different instances of the messages, which are initiated by different nodes, could be merged (or even fused with additional information) if they report the same event. This requires the nodes to process Class C messages in the Application layer, and to be able to determine that two events announced by different nodes are actually the same. Geocasting is a typical approach to Class C protocols as the information to be disseminated will likely be of interest in a certain geographical area only. In comparison to Class B protocols we use geographical positions for forwarding decisions.

4) Class D: This class consists of protocols that broadcast information about non-urgent events whose expiration time is much longer than those of Class C events. The detection rate of a Class D event, i.e., the number of detecting nodes per second, is typically smaller than in Class C. However, because of the much longer expiration time of a Class D event, such an event can still be detected by a large number of nodes. Thus, it is important to identify and merge announcements by different nodes for the same event. Because Class D events are of lower priority than events in the other classes, our proposed Class D protocols are based on distributed caching and their bandwidth consumption is adapted to the available bandwidth. The dissemination scope is similar to Class C protocols, i.e., the target will be a geographical area. The unique feature of Class D protocols is the ability to merge different reports according to their content and relevance.

C. Mapping Broadcast Classes for Building Applications

In Table II, we list possible VANET applications for *Safety, Traffic Efficiency, and Infotainment Applications* as well as Network Coordination [1], [3] and outline how they can be mapped to our class-based forwarding model. The proposed assignment of classes is not engraved in stone; we choose it to demonstrate the classification idea. The general idea is that one or multiple broadcast classes together build the basis for the application needs. Certainly, this can be further extended assuming a multi-technology approach by integrating for example LTE, Wi-Fi, Bluetooth, and others. This is known as *heterogeneous vehicular networking* [4].

Besides the mapping of applications to broadcast classes, it is important to highlight the dependencies between the classes. The dependency between the various protocols can be demonstrated by the following example: Suppose that a vehicle makes an emergency stop due to some road hazard. It informs its 1-hop and 2-hop wireless neighbors of this stop using a Class B protocol. As shown later, the information about these neighbors is available from the routine execution of a Class A protocol. All the nodes that are informed of the emergency stop invoke a Class C protocol and inform further nodes, in a radius of $1\,\mathrm{km}$ say, about a traffic jam. The nodes that become aware of the traffic jam create a Class D event, and disseminate this event to distant nodes.

Figure 2 shows a simplified design overview of our class-based broadcasting approach. Four distinct protocols are exchanging information over the wireless channel, each using a different priority. The priorities defined for each class (see Table I) are mapped to MAC layer Access Categoriess (ACs), as defined in the IEEE 802.11p protocol. This mapping helps to enable a completely self-organizing interaction between all protocol classes, thus prioritizing Class B over Class C and Class D, as well as prioritizing Class C over Class D.

TABLE II
FITTING OF VANET APPLICATIONS WITH CLASS-BASED BROADCAST

Application	Class A	B	C	D
Safety Applications				
Cooperative awareness	✓			
Intersection collision warning	✓	✓		
Lane change assistance (blind spot)		✓		
Overtaking vehicle warning		✓	✓	
Head on collision warning (frontal)		✓		
Rear end collision warning		✓	✓	
Cooperative forward collision warning	✓	✓		
Emergency vehicle warning		✓	✓	
Pre-crash sensing / warning	✓	✓		
Co-operative merging assistance		✓		
Emergency electronic brake lights		✓	✓	
Wrong way driver warning	✓	✓	✓	✓
Stationary vehicle warning	✓	✓	✓	✓
Traffic condition warning		✓	✓	✓
Signal violation warning		✓	✓	
Collision risk warning		✓	✓	
Hazardous location notification		✓	✓	✓
Control loss warning	✓	✓		
Traffic Efficiency and Management Applications				
Green light speed advisory			✓	✓
Platooning	✓	✓		
Cooperative navigation / TIS				✓
Parking space information			✓	✓
Public transport lane		✓	✓	
Infotainment Applications				
POI information				✓
Local advertisements				✓
Media downloading				✓
Multi-player games				✓
Network Coordination				
Neighborship management	✓			
Multi-channel multi-radio coordination	✓	✓		

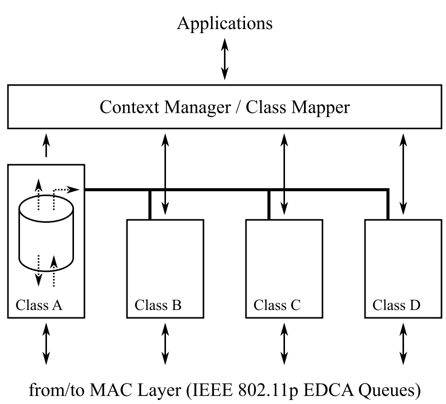


Fig. 2. System design overview. A context-aware class mapper coordinates data exchange of applications with four distinct protocols, each connected to the MAC layer using different priorities and each fulfilling a specific role.

New data is generated locally (for Class A) or received from the Application layer (Class B, Class C, and Class D). Passing the right information to the right protocol is handled by a context-aware class mapper. The Class A protocol maintains a neighbor table and provides access to its contents to other protocols, e.g., for forwarding decisions. This allows each individual protocol to operate autonomously from the Application layer, e.g., for relaying messages. The context manager is responsible to classify data received by the Application layer and can convert messages between classes depending on metrics outlined in Section III-A, e.g, Class B to Class C.

The interaction between the classes is not limited to the information management by the class mapper and the neighbor data collected by Class A protocols. Competition of each class for channel access is handled by the Enhanced Distributed Channel Access (EDCA) mechanism defined in the IEEE 802.11 protocol suite. We prioritize messages (cf. Table I) by assigning them to the corresponding EDCA access categories. This guarantees prioritization and provides a sufficient level of fairness to prevent starvation of flows.

The key selling point of this protocol architecture is that we separate Network layer functionality and application logic. The advantage is that *(a)* the message forwarding is transparent to the applications and *(b)* redundant information by applications concurrently accessing the wireless channel can be avoided.

IV. DETAILED DESIGN OF THE VARIOUS PROTOCOLS

In this section, we propose a detailed design of specific protocols for the various classes, based on the principles introduced in Section III. Our purpose is to demonstrate how the various requirements of each class can be addressed by specific protocols, and how these protocols differ, although all are considered "broadcast protocols." We build upon existing proposals in standardization and in the scientific literature, but also introduce completely new protocols or substantial changes for our integrated class-based broadcasting architecture.

A. Class A Protocols

This class consists of protocols that broadcast routine periodic (beacon) messages to nearby vehicles; primarily to inform every node about the presence of a car. While existing protocols have been designed to allow each node to inform its 1-hop neighbors of its existence, we believe that a more sophisticated protocol is necessary, mainly because the information disseminated by this protocol is needed for the other classes. Specifically, the Class B, Class C, and Class D protocols we propose later on need to know the identities of the 2-hop neighbors of each node. A naïve implementation is that each node v includes the identities of its 1-hop neighbors in its beacons. This will greatly increase the length of these messages in urban environments, and increase their MAC collision probability.

To avoid this problem, we use a Bloom filter [27] (see Appendix VI for more details) in the following way: For each neighbor w of node v, node v adds the node ID of w (e.g., its MAC address) to a local array. This neighbor information is stored in a set $\mathbb{T}_v$, which thus represents all 1-hop neighbors of node v. Moreover, each node v maintains a Bloom filter $\mathcal{T}_v$ storing all entries of $\mathbb{T}_v$, i.e., $\mathcal{T}_v \leftarrow \mathbb{T}_v$. This Bloom filter is added to the beacon messages of node v. Conversely, on receiving a Bloom filter $\mathcal{T}_v$, any node can check with high probability whether a local neighbor is also a neighbor of v. Class A maintains a neighbor table where the information of all the 1-hop neighbors is stored, including their position, and their Bloom filter. A timer expires old entries in the neighbor table, whenever two consecutive beacons have not been received. When node u receives the beacon from v, it also updates its information about its 2-hop neighbors that can be accessed via v. All 2-hop neighbors are then collected in form of a Bloom filter as $\mathcal{T}'' = \bigcup_{v \in \mathbb{T}} \mathcal{T}_v$. For this, the union of two Bloom filters can easily be calculated by performing a logical OR operation on each bit field of the two Bloom filters [43].

In summary, our Class A protocol maintains an exact list of 1-hop neighbors with their respective geographical position as well as a probabilistic list of 2-hop neighbors in form of a Bloom filter. We will discuss the use of this information within the Class B, Class C, and Class D protocols.

B. Class B Protocols

Recall that this class consists of protocols that broadcast information about an emergency event that is likely to be detected only by a few vehicles. The information needs to be broadcast to all N-hop neighbors of the originating node, where N is determined by the application and is typically 1 or 2. The value of N is added to a TTL-like field in the Network layer header of the protocol. It allows every receiving node to make a fast decision, in the Network layer, whether to forward the message or (if the value of this field is 0) to drop it.

In order to mitigate the broadcast storm problem [17], we follow a sender-based rebroadcast decision, where the origin of the broadcast selects the rebroadcast nodes according to the information provided by Class A. The origin node annotates the IDs of the selected rebroadcast nodes to the message together with an individual offset to avoid collisions due to simultaneous rebroadcasts. For $N = 1$, the protocol is very similar to the Class A protocol because relay nodes are not used. We now address the case where $N = 2$, and note that the proposed protocol can be extended for any $N > 2$. Going beyond $N = 2$ requires a change of the protocol behavior not only to add additional nodes in the Bloom filter, which is trivial, but particularly a mechanism to de-synchronize the beacon messages of nodes in a multi-hop vicinity to avoid synchronized collisions. In essence, this leads to a new beacon protocol (Class A), which is beyond the scope of this paper.

In order to reduce the load on the wireless channel, often a greedy approach is used to gain most progress in distance. Particularly to cover non-regular 2D environments (e.g., urban or inner cities), we follow a new approach by choosing as many nodes as needed to cover all our 2-hop neighbors. We take advantage of the idea proposed in [44], where every node v that receives or generates a packet to be broadcast nominates one or multiple of its 1-hop neighbors to rebroadcast the packet. To achieve this goal, node v uses the neighbor information and the Bloom filter sets maintained by the Class A beacon protocol. In detail, node v desires to choose a set of re-broadcasters $\mathbb{R}$ as the minimum subset of 1-hop neighbors in $\mathbb{T}_v$ to cover all of its 2-hop neighbors.

This combinatorial optimization is called minimum set cover problem, which is known to be *NP*-hard [45]. Therefore, we use a greedy iterative process where node v chooses the set $\mathbb{R}$ by selecting a node u that has most new (still uncovered) neighbors and has not yet been selected as a rebroadcast node. Formally, node v starts with an empty set of re-broadcasters $\mathbb{R}$ and a Bloom filter of already-covered 2-hop neighbors $\widehat{\mathcal{T}}''$, which is initialized to the 1-hop neighbors, i.e., $\widehat{\mathcal{T}}'' \leftarrow \mathbb{T}$. It repeatedly chooses the best 1-hop neighbor u as

$$u = \underset{u \in \mathbb{T}}{\arg\max} \left(\mathrm{diff}(\widehat{\mathcal{T}}'', \mathcal{T}_u) \right)$$

and adds this node u to $\mathbb{R}$ and its Bloom filter $\mathcal{T}_u$ to $\widehat{\mathcal{T}}''$.

The quality of a specific Bloom filter to estimate $\mathrm{diff}(\mathcal{A}, \mathcal{B})$, the number of entries in a Bloom filter $\mathcal{B} \leftarrow \mathbb{B}$ that are not part of a local filter $\mathcal{A}$, can formally be expressed as $\mathrm{diff}(\mathcal{A}, \mathcal{B}) = |\mathcal{A} \cup \mathcal{B}| - |\mathcal{A}|$. Here, the estimation of the cardinality [46] of the two Bloom filters can be approximated as

$$|\mathbb{B}| \approx |\mathcal{B}| = -\frac{m \ln(1 - \frac{\mathrm{c}(\mathcal{B})}{m})}{k}.$$

The function $\mathrm{c}(\cdot)$ counts the number of bits set to 1 within the Bloom filter. In the special case of (almost) all bits set to 1 in a Bloom filter, the cardinality is not defined.

The process ends when all 2-hop neighbors are covered, as can be derived by comparing $\widehat{\mathcal{T}}''$ and $\mathcal{T}''$. The set $\mathbb{R}$ now contains all 1-hop neighbors selected to rebroadcast the message. Since $\mathbb{R}$ is usually small (e.g., it is close to 2 in freeway scenarios), it is added to the broadcast message. To avoid a broadcast storm [17], a node receiving this message chooses its rebroadcast delay based on its index i in $\mathbb{R}$ as $i \times t_{\text{rebroadcast}}$. If the cardinality of $\mathbb{R}$ is too large to include all chosen 1-hop neighbors, the addresses of these nodes could also be replaced by a Bloom filter $\mathcal{R} \leftarrow \mathbb{R}$. To increase reception reliability duplicates of each Class B message could be sent.

C. Class C Protocols

Our Class C protocol uses geo-routing and reliable broadcasts to disseminate information about a certain, detected event within a specific geographic region. A message of this type consists of a destination (currently a simple geo-position) where the information should be propagated to, and a lifetime. While in our Class B protocol a decision whether to forward a message is made in the Network layer, using information that appears in the Network layer header, in our Class C protocol the decision is made in the Application layer. This layer reads the information about the event and makes a forwarding decision based on the location of the detecting node, the event time, and whether a report about a similar event has already been received. Our proposed framework for Class C protocols works as follows.

1) First, the node decides whether to report the detected event. This is done in a probabilistic manner, based on the number of detecting nodes, the event type, and the number of 1-hop or 2-hop neighbors of v.
2) Second, the node determines the destination of the event. This could be a geo-position or even a geographical area.
3) Third, the node forwards the event towards its destination. In general, every node that receives an event determines whether it is already acknowledged (that is,

a duplicate to be dropped) and whether to forward the event based on its 1-hop neighbors as provided by Class A. We decided to piggyback the ACK to the rebroadcast to reduce congestion.

In the following, we outline an example algorithm for selecting an appropriate set of forwarders. It fits the case where the originator v needs to broadcast the information to all the nodes between v and the destination geo-position on a one dimensional highway. In this case, v nominates another node u that will nominate another node w and so on. In general, we prefer to nominate a node in the direction of the broadcast, but if this is not possible, we perform store-carry-forward until the Class A protocol provides a fitting neighbor. The algorithm can be extended to 2D setups by invoking a separate instance of the protocol for each direction of the broadcast. The operation of our event forwarding is outlined in Algorithm 1.

Algorithm 1 Class C Event Forwarding for Node v

Input: e, the event to be forwarded

$\mathbb{B} \leftarrow \varnothing$
while no Application-layer ACK received $\wedge$ retransmit limit not reached **do**
  $\mathbb{F} \leftarrow \{n \in \mathbb{T} : n \text{ is towards the destination of } e\} \setminus \mathbb{B}$
  if $\mathbb{F} \neq \varnothing$ **then**
    $u \leftarrow \arg\max_{n \in \mathbb{F}} (\text{distance}\,(n, v))$
    $m \leftarrow$ createMessage(e, u)
    broadcast(m) with delay $\mathcal{U}(0, t_{\text{rebroadcast}})$
    $\mathbb{B} \leftarrow \mathbb{B} \cup \{u\}$
  else
    store-carry-forward
  end if
end while

D. Class D Protocols

The Class D protocols disseminate information about non-urgent events, such as reporting an available parking lot or a traffic jam in a distant area (recall that a traffic jam in a nearby area, which is a much more time-sensitive event, is reported by a Class C protocol). The lifetime of a Class D event is up to a few minutes, much longer than that of Class B and Class C events. During this time period, the event is likely to be detected by many vehicles.

The approach we propose for Class D is based on the following concepts. Information-centric forwarding: information is processed in the Application layer, and it can be aggregated, modified, or invalidated before being forwarded to other vehicles. Store-carry-forward: a moving vehicle carries the information until it meets a new vehicle with which it shares this information. Spatio-temporal forwarding: the decision whether to forward a piece of information on a particular event depends on the time and place it was triggered.

Class D protocols maintain a knowledge base consisting of entries with geographic constraints and their expiration time. Based on these parameters, a broadcast decision can be taken. We decided to build this protocol upon ATB [19], which already supports the management of knowledge bases, message prioritization, and channel quality estimation for congestion aware channel access:

1) A node u that detects a Class D event, or receives a message about such an event, adds or merges it into its database (cache) of active events.
2) Every node u determines the rate of new neighbors ρ detected by its Class A protocol. Whenever u detects a new neighbor v, it makes a probabilistic decision with $p = \frac{1}{\rho}$ whether to inform v about events in its cache.
3) If u decided to inform v about its relevant stored events, u transmits a digest including fingerprints of all available events in the knowledge base. Node v responds with an event request according to Algorithm 2.

Algorithm 2 Class D Event Request for Node v

Input: $\mathbb{D}$, the received digest from node u

$\mathbb{E} \leftarrow \varnothing$
for $d \in \mathbb{D}$ **do**
  if distance$(v, d_{\text{dst}}) <$ distance$(u, d_{\text{dst}}) \vee$ v is driving towards d_{dst} **then**
    $\mathbb{E} \leftarrow \mathbb{E} \cup \{d\}$
  end if
end for
$m \leftarrow$ createMessage$(\mathbb{E}, u)$
broadcast(m)

4) After node u receives the event request from v, it constructs and broadcast a message containing all missed information – limited by the maximum packet size.
5) Let ϕ be the actual size of 1-hop neighbors reported by Class A. When node y receives new information, it informs every node from its 1-hop neighbors with probability $p = \frac{1}{\phi}$ about any relevant Class D events it has in its knowledge base.
6) Every node z periodically checks its cache and prunes obsolete events: any event whose expiration time has arrived or any event that is not relevant to the current location of the node.

Moreover the overall concept can easily be extended to include additional metrics for p: the due date of the event (when the due date is closer, the probability is smaller); the area to which this event is relevant (when the node moves closer to the border of this area, the probability decreases); how busy the wireless channel is (when the channel is busier, the probability is smaller).

Note that typical protocols described in the literature fulfilling similar tasks are based on unicast [35], [47]. We, however, determined that unicast may lead to substantial performance issues in this application domain [8].

V. Performance Evaluation

We evaluated our class-based broadcasting approach to demonstrate its feasibility as well as to gain more insights into the resulting performance gains. Our main focus is to underline the need for different broadcast protocols in accordance with the selected application requirements. Due to space restrictions, we only report on results for selected protocol configurations.

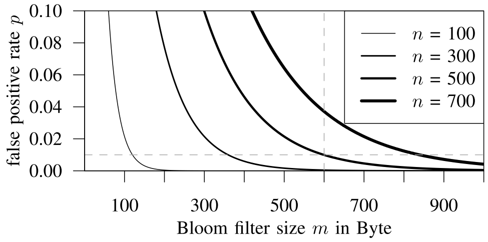


Fig. 3. False positive rate p of a Bloom filter as a function of its length m and the inserted element count n.

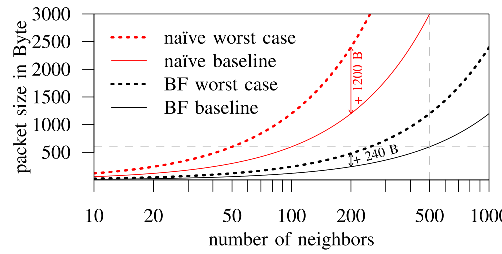


Fig. 4. Packet size for neighbor information as a function of the number of neighbors; for the Bloom filter (BF) approach we assume $p = 0.01$, and for the naïve approach we use 6 Byte per neighbor entry.

A. Optimal Bloom Filter Size

The optimal number of hash functions k to minimize the false positive error rate for a given Bloom filter of size m and inserted element count n can be derived [43] as

$$k \approx \frac{m}{n} \ln 2 . \tag{1}$$

Similarly, for a Bloom filter of length m bits and n inserted elements, the false positive error rate [28] for testing whether an element is member of this Bloom filter can be obtained as

$$\mathrm{p}(m, n) = \left[1 - \left(1 - \frac{1}{m}\right)^{kn}\right]^{k}, \tag{2}$$

where k is calculated by Equation (1).

To estimate the best Bloom filter size for our application scenario, we consider an element count derived from empirical evaluations as follows. In Figure 3, we plot the false positive rate as a function of the Bloom filter size m and inserted element count n. Our simulation results for the neighbor table experiments show a maximum number of 500 1-hop neighbors for each vehicle in the high density scenario. Thus, assuming a maximum false positive rate of 1 %, a Bloom filter of 600 Byte perfectly matches. It also nicely fits into a CAM message of up to 800 Byte, as used by the Class A protocols. We explicitly note that the Bloom filter size can be chosen according to the needed application demands and does not limit the amount of neighbors that can be inserted to it.

The evaluation so far concerned the storage of IDs of cars in a Bloom filter structure assuming that this ID is fixed and not changed over time. Privacy preserving schemes, however, suggest the use of so-called temporary pseudonyms for use in VANETs. The general idea is to continuously change the ID of the car to a new pseudonym (or even swap it with another nearby car) in order to introduce entropy and to disallow tracking of the car's routes. A wide range of pseudonym handling schemes has been proposed since [48]. These ideas have also been adopted by the standardization bodies and the current ETSI ITS-G5 standard recommends to change IDs frequently [49].

In the following, we investigate the impact of such ID changes on the Bloom filter size and compare it to the naïve approach using complete IDs for the exchanged neighbor tables. In particular, we investigate the size of neighbor information when using a Bloom filter with a false positive rate of $p = 0.01$, and a naïve approach (neighbor entries sent as a plain list) where each neighbor entry takes 6 Byte of payload. Further, we assume that in the worst case each vehicle changes its identifier (or pseudonym) for each sent beacon, which leads to an increase of the number of elements in the neighbor table by the factor of two if entries are outdated after missing two consecutive beacons. For this, we repeated the simulations on the Bloom filter size. As shown in Figure 4, the increase of the packet length using the naïve approach is 1200 Byte for a neighbor count of 200. However, our Bloom filter approach takes only 240 Byte of additional payload for the same scenario. We conclude that the Bloom filter is a very appropriate data structure even in case of implemented privacy preserving techniques.

B. Simulation Setup and Metrics

For all simulations, we used the *de facto* standard for vehicular networking simulation, Veins [50], which couples the SUMO road traffic mobility simulator with the network simulator OMNeT++.

Veins is well established and widely used in the vehicular networking community and provides an extensive suite of models for vehicular communication, each having been closely validated against measurements done in extensive Field Operational Tests (FOTs). Among these validated models are channel models that can capture obstacle shadowing [51], two ray fading [52], [53], antenna radiation characteristics [54], and adjacent channel interference [55] as well as Physical layer and MAC layer models of ARIB T109 [56] and IEEE 802.11p [8]. The packet error rate model in our simulation is based on the NIST error rate model of ns-3 [57], [58].

For our simulation setup we configured *(a)* a 7 km freeway scenario and *(b)* a 9 km road segment of a Manhattan scenario, respectively. For the latter, we looked at one of the major avenues (such as 5$^{\text{th}}$ Avenue in Manhattan downtown) including the simulation of all cross traffic. In order to mitigate potential border effects, we configured Veins to perform the network simulation only within a region centered at the middle of the respective scenario. We further configured a Region of Interest (ROI) in which we collect protocol performance metrics, cf. Table III.

Road traffic for the freeway scenario was modeled in SUMO by sampling from a distribution of five different vehicle types (two types of trucks and three types of cars modeling a variety of driving styles). For the Manhattan scenario no trucks were used and four types of vehicles modeling a variety of driving styles were used. We chose SUMO in favor to vehicular traces as it gives us the ability to better evaluate protocol behavior by using different vehicle densities and mobility patterns.

We use a warm-up period of 289 s (freeway) and 59 s (Manhattan) for SUMO to fill the scenario with vehicles and

TABLE III
SIMULATION PARAMETERS

SUMO simulation setup	
Freeway length: SUMO, OMNeT++, ROI	7, 5, 3 km
Manhattan length: SUMO, OMNeT++, ROI	9, 7, 5 km
Vehicle density (freeway): low, high	∼ 43, ∼148 veh/km
Vehicle density (Manhattan): low, high	∼ 56, ∼207 veh/km
Number of lanes	6 (3 per direction)
Percentage of cars and trucks (freeway)	90 %, 10 %
ETSI ITS-G5 TRC	
Min/default/max interval $I_{\min}$, I_{def}, $I_{\max}$	40 ms, 500 ms, 1 s
Channel busy fraction thresholds $b_{\min}$, $b_{\max}$	0.15, 0.40
ATB	
Beacon interval range $I_{\min}$, $I_{\max}$	100 ms, 1 s
Channel/interval weighting w_{C}, w_{I}	2, 0.75
IEEE 802.11p MAC	
Packet size Class A	800 B
Class A Bloom filter size	600 B
Packet size Class B and C	300 B
Packet size Class D digest	each 8 B, max. 1024 B
Packet size Class D KB entry	each 64 B, max. 1024 B
MAC priority Class A	AC_BE
MAC priority Class B	AC_VO
MAC priority Class C	AC_VI
MAC priority Class D	AC_BK
IEEE 802.11p PHY	
NIC bitrate	6 Mbit/s
NIC TX power	20 mW
NIC sensitivity	−89 dBm
Frequency	5.89 GHz
Path loss model	freespace ($\alpha = 2.0$)

TABLE IV
PROTOCOL PARAMETERS

Parameter	Class B	Class C	Class D
triggered nodes	10, 25, 50	1	2
trigger f	10 Hz	{10 Hz, 4 Hz, 2 Hz}	
$t_{\mathrm{rebroadcast}}$	25 ms	25 ms	-
duplicates	2	-	-
retransmits	-	3	-

an additional warm-up period of 11 s for OMNeT++ to reach a steady state of Class A protocol operations and to populate 1-hop and 2-hop neighbor tables. Moreover, in this 11 s warm-up period we pre-populate the knowledge base of vehicles with information items. Only after this time we invoke Class B, C, and D protocols and the recording of results. All relevant simulation parameters are summarized in Tables III and IV.

In order to investigate our class-based broadcasting architecture, we study the performance in multiple dimensions. Classical metrics for wireless networking in the vehicular context, such as the packet success rate and the channel utilization, provide little insight into the behavior of the respective vehicular networking applications [1]. Therefore, we primarily looked at Application layer class-based metrics.

We further want to comment on explicit comparison to the state of the art. For Class A, we use protocols as presented in the literature and extend these for our Bloom filter based approach. Similarly, we use existing concepts for the Geonetworking solutions. The main emphasis of our paper is to make all these protocols being able to co-exist. In the experiments, we therefore concentrate on this part following a stepwise approach starting with Class A, then integrate Class B, and so on.

For Class A, we first investigated the beacon interval to gather insights on the latency of new status messages. Besides providing vehicle status updates via CAM messages, our Class A protocols also maintain the 1-hop and 2-hop neighbor tables. To assess the quality of the neighbor tables, i.e., the up-to-dateness of 1-hop and 2-hop entries, we compare each Class A beacon protocol against an oracle. The oracle calculates the neighbor set according to a unit-disk model. For the distance of nodes to be treated as 1-hop neighbors we use the 99 % quantile of 1-hop distances of our Class A sample experiments for the communication range. This allows us to derive two sets of neighbors as Bloom filters: that estimated by the Class A protocol $\mathbb{T}$ and that calculated by the oracle $\mathbb{O}$. Based on $\mathbb{T}$ and $\mathbb{O}$, we use the following two metrics to evaluate the quality of neighbor tables: The fraction of missing neighbors of a node compared to the oracle is calculated as $\frac{|\mathbb{O}\setminus\mathbb{T}|}{|\mathbb{O}|}$. The fraction of outdated neighbors describes the relative amount of superfluous neighbors compared to the oracle as $\frac{|\mathbb{T}\setminus\mathbb{O}|}{|\mathbb{T}|}$. These metrics are observed every 100 ms after the warm-up period; for the 2-hop neighbors similar metrics are measured taking advantage of the Bloom filters. For Class B, C, and D protocols, we study two metrics: *(a)* the delay between the observation of an event or the creation of a message that informs of a new event and the time this message has been received by all target nodes; *(b)* the fraction of successfully informed nodes, which can be viewed as an indicator of the reliability of the protocol.

In addition to the above metrics, we explicitly measure the dissemination speed in $\mathrm{m\,s^{-1}}$ for Class D, which indicates how quickly the protocol transports the message through the vehicular network.

For all simulation experiments we performed at least 10 runs with different random seeds for simulating road and network traffic to obtain statistically significant results. In all plots we report the mean value of the selected metric together with its 95 % confidence interval, obtained for all vehicles in all runs.

C. Baseline Experiments

At first, we measure the protocols' individual performance, by running Class A alone as well as in combination with Class B, C, and D at the same time (the latter depend on the neighbor tables established by Class A).

1) Class A Performance: We start with the Class A protocols, which build the foundation for all other broadcast classes because they provide neighbor information. A typical example application is cooperative awareness (cf. Table II).

In Figure 5a, we plot the mean beacon intervals. The results for the 1 Hz and 10 Hz protocol options are trivial; i.e., they are 1000 ms and 100 ms respectively. TRC's beacon intervals oscillate between the different protocol states. As the wireless channel becomes congested, TRC converges to an average delay of 500 ms. ATB continuously uses smaller beacon intervals. We also explore the success rate and channel utilization. The 10 Hz protocol massively overloads the chan-

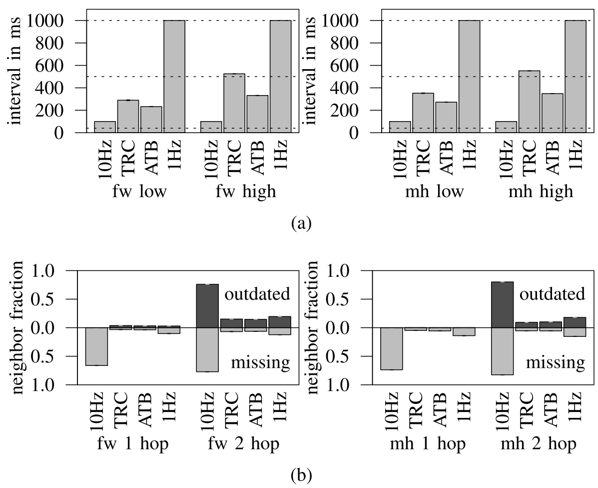


Fig. 5. Class A protocol performance for the freeway and Manhattan scenario and different vehicle densities. (a) Resulting beacon interval indicating the latency performance. (b) Neighbor ratio for 1-hop and 2-hop neighbors (high density scenarios).

nel whereas all other protocols carefully control the channel utilization (data not shown due to length constraints).

Most importantly, we investigate the number of outdated and missing entries in the neighborship data and compared our results to those of an oracle. Figure 5b shows the results for the high density scenarios. As can be seen for 1-hop-neighbors, the fraction of missing entries is extremely high (about 60 %) for the 10 Hz beaconing; missing entries are those that have been identified by the oracle but not the protocol under observation. This is due to packet collisions and, thus, not received neighbor information. All other protocol options perform better, particularly TRC and ATB. We can also observe the impact of the mobility on the Class A protocols: In the Manhattan scenario the amount of outdated 1-hop neighbors is lower than on the freeway scenario due to slower driving speeds. The results for 2-hop neighbor information are similar, but the amount of outdated neighbors is higher since dissemination time accumulates over 2 hops; outdated entries are those that should have been pruned, again, according to the oracle. In this case, also a high number of outdated entries can be observed. This is due to many packet collisions: beacons are lost and direct 1-hop neighbors may be reported as 2-hop neighbors by another direct neighbor. TRC and ATB perform best with respect to this metric.

Due to the inability of the 10 Hz protocol to provide accurate neighborship information, we only report results for TRC, ATB, and 1 Hz in the following.

2) Class B Performance: Next, we study the performance of our Bloom filter based Class B protocol. Recall that Class B protocols provide urgent information in an N-hop range as needed by many safety related applications (e.g., intersection collision warning). We configured $N = 2$ for the following experiments. We are primarily interested in the resulting delay and the fraction of nodes that successfully received the broadcasts. We study an increasing number of selected broadcast initiators, i.e., increasing load on the wireless channel. Figure 6a shows the observed delays in the Manhattan scenarios. We only plot results for the experiments using TRC as the Class A protocol; 1 Hz beaconing led to similar results as well as for the freeway scenario. For ATB (not shown due to space constraints) we observe a slightly higher delay caused by the lower beaconing interval. The key insight we gain is that the delay primarily depends on the load on the wireless channel. If we either switch from low density to high density or from a few selected broadcast initiators to a larger number, the delay increases from about 70 ms to more than 150 ms.

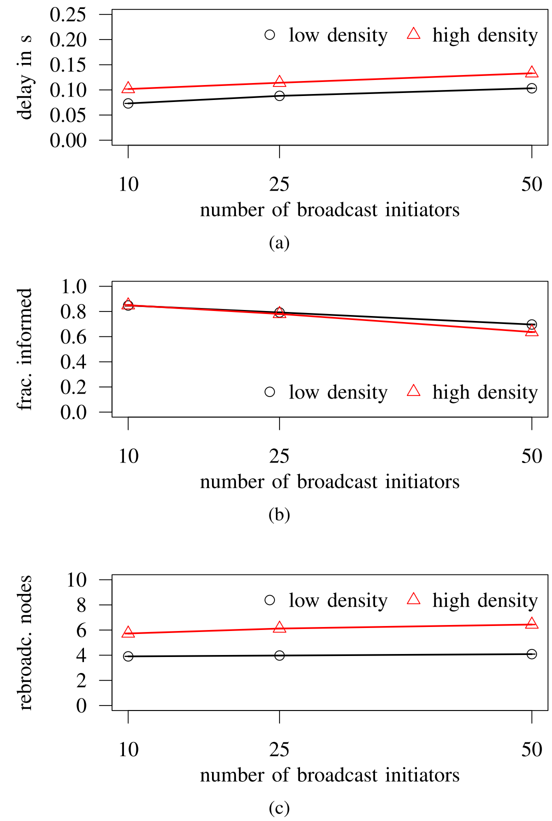


Fig. 6. Class B protocol performance for different numbers of broadcast initiators. Plotted are the results using TRC as the Class A protocol and the Manhattan scenario; results are similar for the freeway scenario. (a) Delay for low and high density Manhattan scenarios. (b) Fraction of successfully informed nodes for low and high density Manhattan scenarios. (c) Selected rebroadcast nodes for low and high density Manhattan scenarios.

Figure 6b shows the fraction of informed nodes as a function of the number of broadcast initiators. We see again a significant difference between the low and high density scenarios. When the channel load reaches saturation, the success rate shows a decreasing trend. Still, more than 60 % of the vehicles can be informed in the worst case.

In Figure 6c, we show the number of selected rebroadcast nodes of our Bloom filter based approach, that is, the number of forwarders such that all 2-hop neighbors can be informed. We see that with increasing vehicle density and number of broadcast initiators the number of selected rebroadcast nodes increases as well, which is caused by the now substantial network load.

To show the advantage of our Bloom filter based rebroadcast protocol, we compare it against a classical greedy approach (as used, e.g., in DV-CAST [59]) where we select two rebroadcast nodes from a node's neighbor table, namely the leftmost and rightmost neighbor. In Figure 7, we show the fraction

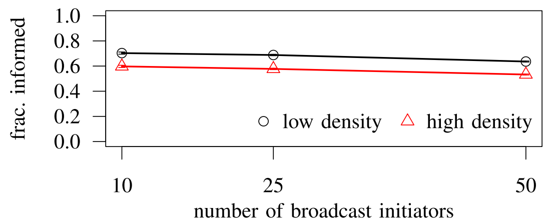
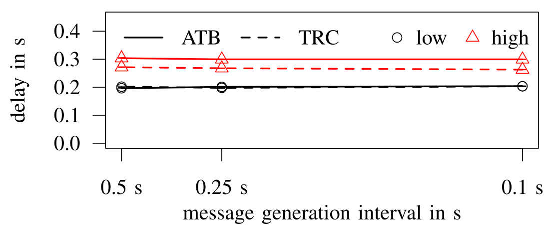
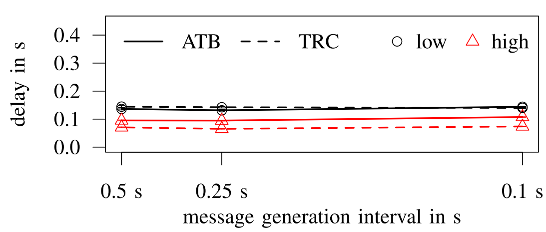


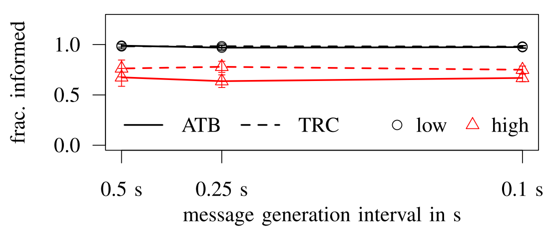
Fig. 7. Fraction of successfully informed nodes for low and high density Manhattan scenarios using a greedy approach for message dissemination. Plotted are the results using TRC as the Class A protocol and the Manhattan scenario.



(a)



(b)



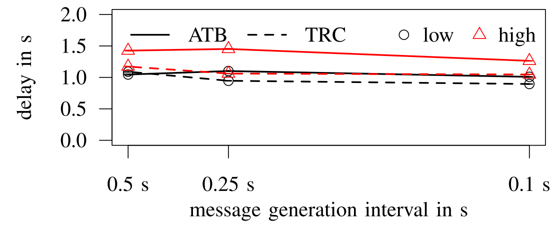
(c)

Fig. 8. Class C protocol performance for low and high density scenarios and different message generation intervals. Plotted are the results using the ATB and TRC Class A protocols; the results are comparable to 1 Hz beaconing. (a) Delay for Manhattan scenarios. (b) Delay for freeway scenarios. (c) Fraction of successfully informed nodes for Manhattan scenarios.

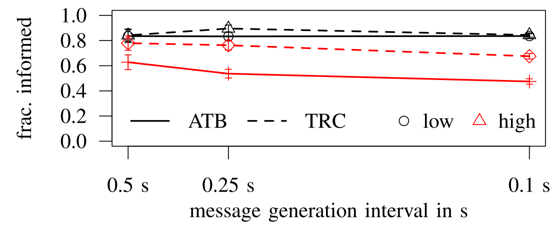
of informed nodes for different numbers of broadcast initiators. Compared to our Bloom filter approach in Figure 6b, we observe a lower number of informed nodes. We conclude that the Bloom filter based solution not only helps in neighbor management in Class A protocols but also for 2-hop data dissemination in larger networks.

3) Class C Performance: We performed the same experiments for Class C protocols for the resulting delay and the fraction of nodes that successfully received the message. An example application is an emergency vehicle warning system (cf. Table II), which disseminates messages along the road. We study the protocol behavior for decreasing message generation intervals, i.e., a slowly increasing network load.

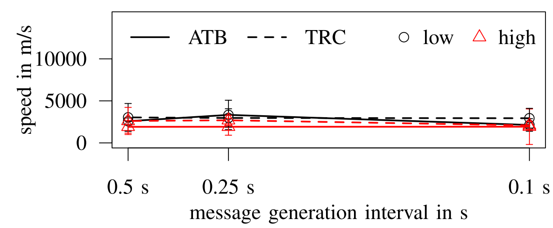
As can be seen in Figure 8a for the Manhattan scenario, the delay for Class C protocols depends on two key factors. First, the channel load is important: the higher the traffic density, the more the channel becomes loaded, which



(a)



(b)



(c)

Fig. 9. Class D Performance for different message generation intervals in low and high density scenarios. Plotted are the results using the ATB and TRC Class A protocols; results for 1 Hz are comparable to TRC; results are similar for the freeway scenario. (a) Delay for Manhattan scenarios. (b) Fraction of successfully informed nodes for Manhattan scenarios. (c) Dissemination speed of new messages for Manhattan scenarios.

translates to greater message delays. Secondly, the availability of rebroadcasters, namely, nodes that are able to forward the message, plays a significant role. On the other hand the more up-to-date the neighbor tables are and the more likely it is to have sufficient rebroadcasters available in the direction of the geocast, the lower delays can be observed. This can be observed in the freeway scenario in Figure 8b where the delay in the high density scenario is lower than in the low density.

However, the delay is only partly telling the story. As can be seen in Figure 8c, which shows the fraction of informed nodes, with a reduced channel load the fraction of successfully informed nodes becomes larger. For the freeway scenario (data not shown due to length constraints) the received fraction is in general lower due to the higher mobility, but shows similar qualitative effects.

4) Class D Performance: One possible application using this broadcast class are Traffic Information Systems (TISs). In order to assess the performance of Class D protocols, we periodically selected two vehicles to insert information items into their local knowledge base. Each entry is configured with a destination position of the other selected vehicle's geoposition.

We first look at the resulting delay as plotted in Figure 9a for the Manhattan scenarios. As can be seen, the delay is not very sensitive in the low density scenarios but becomes critical in the high density scenario. We also note the dependency on the underlying Class A protocol. The ATB beaconing in the high density scenario leads to slightly larger delays compared to the TRC protocol. This is due to the higher channel utilization caused by ATB which chooses lower beaconing intervals than TRC. Similar effects are observed in the freeway scenario, as well as for 1 Hz beaconing. This trend can be confirmed when looking at the fraction of informed vehicles in Figure 9b. We notice that our Class D protocol is able to inform more than 80 % of vehicles on the road in the low density scenarios. When road traffic density increases, the Class A beaconing protocol leaves less channel capacity for class D protocols, thus the fraction of informed nodes decrease. This is perfectly in line with our integrated broadcast approach using the EDCA subsystem of IEEE 802.11p MAC, where higher prioritized protocols (Class A) have a higher probability for channel access than lower prioritized protocols (Class D). For 1 Hz beaconing (data not shown due to space restrictions) as well as for the freeway scenario similar trends can be observed.

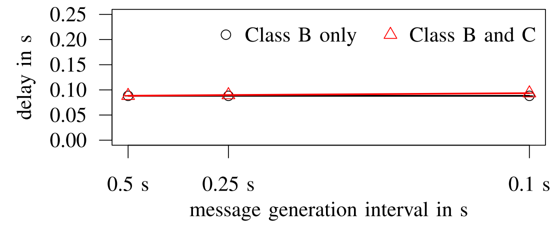
As a final metric, we also investigate the data dissemination speed through the vehicular network. Figure 9c plots the dissemination speed in meters per second for low and high density Manhattan scenarios as well as the different Class A protocol options. As can be seen, the confidence levels are overlapping, thus, no clear winner can be identified. Yet, the speed is in general smaller in the TRC/ATB configuration, particularly when the load in the network is increasing. This is due to the MAC protocol priority of Class D packets receiving the largest contention window and, therefore, enabling the adaptive Class A protocols to successfully compete for more channel capacity. These results perfectly confirm our vision that an integrated class-based broadcasting architecture is important.

D. Studying the Integrated Class Based Broadcast Architecture

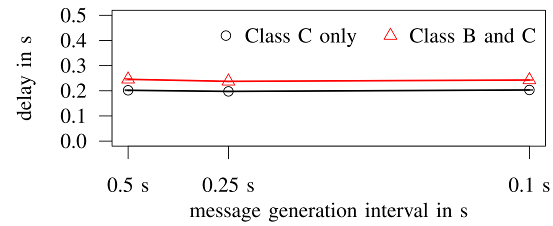
The advantages and capabilities of our novel class-based broadcasting architecture for vehicular networks becomes even more visible when looking at the integrated performance in more holistic experiments. We stepwise enable all protocols and investigate the dependencies of the protocol classes – a procedure that seconds the ultimate need for integrated broadcast protocol classes.

1) Dependencies Between Class B and Class C: We first investigate the dependencies between Class B and Class C protocols. We thus configured a setup enabling Class A for neighborship management as well as cooperative awareness, Class B for emergency or warning messages, e.g., about a just happening traffic accident, as well as Class C for informing other cars about lower priority events, e.g., green light speed advisory (cf. Table II).

Figure 10 shows the results for the low density Manhattan scenario, and using TRC as a Class A protocol. We keep the message rate for Class B constant and vary the data rate of Class C messages. Figure 10a indicates the impact of Class C on Class B with respect to the delay. As can be seen, the delay of Class B messages is not effected at all, which is exactly the expected result. This is mainly accomplished by the EDCA subsystem of the IEEE 802.11p MAC, which gives higher



(a)



(b)

Fig. 10. Integrated Class A, B, and C performance for different message generation intervals; results plotted for using TRC as Class A protocol and a low density Manhattan scenario; results for ATB and 1 Hz are similar to TRC; results are similar for the freeway scenario. (a) Delay of Class B messages. (b) Delay of Class C messages.

priority messages better channel access probabilities. Class B messages are also not affected in terms of the fraction of informed nodes as well as for other protocol options of Class A (not plotted due to space limits).

However, when looking at the resulting performance of our Class C protocol, we see the high impact of the protocol interaction. Due to the increasing channel load caused by Class B, which is working on a higher priority level compared to Class C, the observed delay of Class C messages increases by a factor of about 25 %, as shown in Figure 10b. Interestingly, the fraction of informed nodes for Class C only marginally depends on the concurrently running Class B protocol (not plotted due to space restrictions). The main reason is that there is still sufficient time for the Class C protocol to deliver the messages using geocasting; the delivery is simply delayed. Only in high density scenarios packet loss influences the delivery rate negatively. In particular, in the high density Manhattan scenario and using ATB as Class A protocol we observe slightly lower delays for Class C operation when Class B protocols are enabled, however with significantly lower delivery rates. This is because the channel is overloaded. We observed similar results for the other scenarios and protocol options, but do not show graphs due to lack of space.

2) Dependencies Between Class B/C and Class D: In a similar experiment, we study the interdependencies between Class B and Class D protocols. For this experiment, we configured a setup using Class A for neighborship management and cooperative awareness, Class B for emergency messages e.g., about a lane change, and Class D for non-urgent events such as informing about longer lasting traffic jams on a road network provided by a TIS.

Figure 11 shows the results for the high density Manhattan scenario and using TRC as a Class A protocol. We keep the message rate for Class B constant and vary the data rate of Class D messages. As can be seen, Class D messages have nearly no impact on the Class B protocol. Our Bloom filter

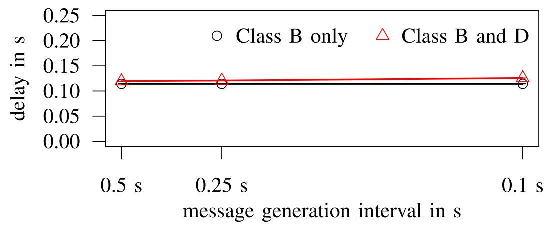


Fig. 11. Integrated Class A, B, and D performance (delay of Class B messages) for different message generation intervals; results plotted for using TRC as a Class A protocol and a high density Manhattan scenario; results for ATB and 1 Hz are comparable to TRC; results are similar for the freeway scenario.

based Class B protocol generates even more messages with an increasing number of cars and when combining this with a second protocol, the channel gets more congested. The fraction of informed cars keeps also almost the same. We already discussed this effect when studying the dependencies between Class B and Class C protocols.

A similar behavior (like Class B with Class D) can be observed when combining Class C with Class D protocols. We therefore omit showing and discussing these results. Possible applications for this scenario are listed in Table II (e.g., rear end collision warning for Class C and TIS using Class D).

VI. Conclusions

In this paper we proposed a novel, integrated, context-aware, broadcast-based Network layer for supporting past and future VANET applications. We also proposed four broadcast classes that match the requirements of all known applications. We showed that these classes not only co-exist on the same Network layer, but also make use of cross-protocol functionality. We analyzed the performance of the proposed protocols and discussed their properties and their ability to co-exist on the same wireless channel. Furthermore, we presented important findings for future protocol designs and evaluated the influence of each protocol when concurrent operation is performed. We see our integrated broadcast protocol approach to provide extensibility and applicability for future protocol designs.

Appendix
Bloom Filter

A Bloom filter is a hash represented in the form of a bit field $\mathcal{B}$ of size m, that is,

$$\mathcal{B} = \{\mathcal{B}_0, \dots, \mathcal{B}_{m-1}\}.$$

We further need a set of k independent hash functions $\{h_0(\cdot), \dots, h_{k-1}(\cdot)\}$, each of which maps its input to a value in $[0, m-1]$, that is, to a bit address. Inserting a node's identifier y into the Bloom filter $\mathcal{B}$ is realized by setting all bits indicated by any of the k hash functions. More formally,

$$\mathcal{B} \leftarrow y \Rightarrow \forall i \in [0 \dots k-1] : \mathcal{B}_{h_i(y)} \leftarrow 1.$$

It is assumed that, initially, all bits in $\mathcal{B}$ are set to 0. Intuitively, this means that the number of 1 bits in the Bloom filter is continuously increasing with each inserted node. Some bits for the new node y' may have been set to 1 already, others are explicitly set to 1.

In order to test whether a node y' is part of the Bloom filter $\mathcal{B}$, the same k hash functions are used to see which bits need to be set. More formally,

$$y' \in \mathcal{B} \Leftrightarrow \forall i \in [0 \dots k-1] : \mathcal{B}_{h_i(y')} = 1.$$

This is a probabilistic test: it can only be said correctly whether a node was *not* part of the Bloom filter. Yet, false positives are possible. For larger m the expected fraction of errors gets smaller. Thus, depending on the expected number of input values and used hash functions, m can be set to a value that keeps the number of false positives small enough for the envisioned application.

References

[1] C. Sommer and F. Dressler, *Vehicular Networking*. Cambridge, U.K.: Cambridge Univ. Press, Nov. 2014.

[2] H. Hartenstein and L. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Commun. Mag.*, vol. 46, no. 6, pp. 164–171, Jun. 2008.

[3] G. Karagiannis *et al.*, "Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 4, pp. 584–616, 4th Quart., 2011.

[4] F. Dressler, H. Hartenstein, O. Altintas, and O. Tonguz, "Inter-vehicle communication: Quo vadis," *IEEE Commun. Mag.*, vol. 52, no. 6, pp. 170–177, Jun. 2014.

[5] *IEEE Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11-2012, 2012.

[6] N. Wisitpongphan, O. K. Tonguz, J. S. Parikh, P. Mudalige, F. Bai, and V. Sadekar, "Broadcast storm mitigation techniques in vehicular ad hoc networks," *IEEE Wireless Commun.*, vol. 14, no. 6, pp. 84–94, Dec. 2007.

[7] J. Jeong, A. Petrescu, T. Oh, D. Liu, and C. Perkins, "Problem statement for IP wireless access in vehicular environments," IETF, Internet-Draft, Tech. Rep. draft-jeong-ipwave-problem-statement-00, Jun. 2017. [Online]. Available: http://tools.ietf.org/html/draft-jeong-ipwave-problem-statement-00

[8] F. Klingler, F. Dressler, and C. Sommer, "IEEE 802.11p unicast considered harmful," in *Proc. 7th IEEE Veh. Netw. Conf. (VNC)*, Kyoto, Japan, Dec. 2015, pp. 76–83.

[9] W. K. Wolterink, G. Heijenk, and G. Karagiannis, "Constrained geocast to support cooperative adaptive cruise control (CACC) merging," in *Proc. 2nd IEEE Veh. Netw. Conf. (VNC)*, Jersey City, NJ, USA, Dec. 2010, pp. 41–48.

[10] S. Oncu, J. Ploeg, N. van de Wouw, and H. Nijmeijer, "Cooperative adaptive cruise control: Network-aware analysis of string stability," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1527–1537, Aug. 2014.

[11] M. Segata *et al.*, "Towards communication strategies for platooning: Simulative and experimental evaluation," *IEEE Trans. Veh. Technol.*, vol. 64, no. 12, pp. 5411–5423, Dec. 2015.

[12] L. Wischhof, A. Ebner, and H. Rohling, "Information dissemination in self-organizing intervehicle networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 1, pp. 90–101, Mar. 2005.

[13] Y.-T. Yu, T. Punihaole, M. Gerla, and M. Sanadidi, "Content routing in the vehicle cloud," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, Orlando, FL, USA, Oct. 2012, pp. 1–6.

[14] M. Sepulcre, J. Mittag, P. Santi, H. Hartenstein, and J. Gozalvez, "Congestion and awareness control in cooperative vehicular systems," *Proc. IEEE*, vol. 99, no. 7, pp. 1260–1279, Jul. 2011.

[15] *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service*, document EN 302 637-2 V0.0.9, ETSI, Nov. 2012.

[16] F. J. Ros, P. M. Ruiz, and I. Stojmenovic, "Reliable and efficient broadcasting in vehicular ad hoc networks," in *Proc. 69th IEEE Veh. Technol. Conf. (VTC-Spring)*, Barcelona, Spain, Apr. 2009, pp. 1–5.

[17] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proc. 5th ACM Int. Conf. Mobile Comput. Netw. (MobiCom)*, Seattle, WA, USA, Aug. 1999, pp. 151–162.

[18] E. Van de Velde and C. Blondia, "Adaptive REACT protocol for emergency applications in vehicular networks," in *Proc. 32nd IEEE Conf. Local Comput. Netw. (LCN)*, Dublin, Ireland, Oct. 2007, pp. 613–619.

[19] C. Sommer, O. K. Tonguz, and F. Dressler, "Traffic information systems: Efficient message dissemination via adaptive beaconing," *IEEE Commun. Mag.*, vol. 49, no. 5, pp. 173–179, May 2011.
[20] R. S. Schwartz, A. E. Ohazulike, C. Sommer, H. Scholten, F. Dressler, and P. Havinga, "On the applicability of fair and adaptive data dissemination in traffic information systems," *Ad Hoc Netw.*, vol. 13, pp. 428–443, Feb. 2014.
[21] G. Caizzone, P. Giacomazzi, L. Musumeci, and G. Verticale, "A power control algorithm with high channel availability for vehicular ad hoc networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Seoul, South Korea, May 2005, pp. 16–20.
[22] T. Tielert, D. Jiang, H. Hartenstein, and L. Delgrossi, "Joint power/rate congestion control optimizing packet reception in vehicle safety communications," in *Proc. 10th ACM Int. Workshop Veh. Internetw. (VANET)*, Taipei, Taiwan, Jun. 2013, pp. 51–60.
[23] *Intelligent Transport Systems (ITS); Decentralized Congestion Control Mechanisms for Intelligent Transport Systems Operating in the 5 GHz Range; Access Layer Part*, document TS 102 687 V1.1.1, ETSI, Jul. 2011.
[24] T. Tielert, D. Jiang, Q. Chen, L. Delgrossi, and H. Hartenstein, "Design methodology and evaluation of rate adaptation based congestion control for vehicle safety communications," in *Proc. 3rd IEEE Veh. Netw. Conf. (VNC)*, Amsterdam, The Netherlands, Nov. 2011, pp. 116–123.
[25] G. Bansal, J. B. Kenney, and C. E. Rohrs, "LIMERIC: A linear adaptive message rate algorithm for DSRC congestion control," *IEEE Trans. Veh. Technol.*, vol. 62, no. 9, pp. 4182–4197, Nov. 2013.
[26] C. Sommer, S. Joerer, M. Segata, O. K. Tonguz, R. L. Cigno, and F. Dressler, "How shadowing hurts vehicular communications and how dynamic beaconing can help," *IEEE Trans. Mobile Comput.*, vol. 14, no. 7, pp. 1411–1421, Jul. 2015.
[27] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Commun. ACM*, vol. 13, no. 7, pp. 422–426, Jul. 1970.
[28] A. Broder and M. Mitzenmacher, "Network applications of bloom filters: A survey," *Internet Math.*, vol. 1, no. 4, pp. 485–509, Jan. 2005.
[29] A. Marandi, M. F. Imani, and K. Salamatian, "Practical bloom filter based epidemic forwarding and congestion control in DTNs: A comparative analysis," *Comput. Commun.*, vol. 48, pp. 98–110, Jul. 2014.
[30] F. Angius, M. Gerla, and G. Pau, "BLOOGO: BLOOm filter based GOssip algorithm for wireless NDN," in *Proc. 13th ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc), 1st ACM Workshop Emerg. Name-Oriented Mobile Netw. Design-Archit., Algorithms, Appl. (NoM)*, Hilton Head, SC, USA, Jun. 2012, pp. 25–30.
[31] A. Marandi, M. F. Imani, and K. Salamatian. (Aug. 2012). "Optimization of bloom filter parameters for practical bloom filter based epidemic forwarding in DTNs." [Online]. Available: https://arxiv.org/abs/1208.3871
[32] K. N. Nakorn, Y. Ji, and K. Rojviboonchai, "Bloom filter for fixed-size beacon in VANET," in *Proc. IEEE 79th Veh. Technol. Conf. (VTC Spring)*, May 2014, pp. 1–5.
[33] F. A. Silva, A. Boukerche, T. R. B. Silva, L. B. Ruiz, and A. A. Loureiro, "Geo-localized content availability in VANETs," *Ad Hoc Netw.*, vol. 36, no. 2, pp. 425–434, Jan. 2015.
[34] D. Borsetti and J. Gozalvez, "Infrastructure-assisted geo-routing for cooperative vehicular networks," in *Proc. 2nd IEEE Veh. Netw. Conf. (VNC)*, Jersey City, NJ, USA, Dec. 2010, pp. 255–262.
[35] *Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 4: Geographical Addressing and Forwarding for Point-to-Point and Point-to-Multipoint Communications; Sub-Part 1: Media-Independent Functionality*, document TS 102 636-4-1 V1.1.1, European Telecommunications Standards Institute, Jun. 2011.
[36] L. Zhang, B. Yu, and J. Pan, "GeoMob: A mobility-aware geocast scheme in metropolitans via taxicabs and buses," in *Proc. 33rd IEEE Conf. Comput. Commun. (INFOCOM)*, Toronto, ON, Canada, Apr. 2014, pp. 1279–1787.
[37] R. Jiang, Y. Zhu, T. He, Y. Liu, and L. M. Ni, "Exploiting trajectory-based coverage for geocast in vehicular networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 12, pp. 3177–3189, Dec. 2014.
[38] F. Farnoud and S. Valaee, "Reliable broadcast of safety messages in vehicular ad hoc networks," in *Proc. 28th IEEE Conf. Comput. Commun. (INFOCOM)*, Rio de Janeiro, Brazil, Apr. 2009, pp. 226–234.
[39] N. Liu, M. Liu, W. Lou, G. Chen, and J. Cao, "PVA in VANETs: Stopped cars are not silent," in *Proc. 30th IEEE Conf. Comput. Commun. (INFOCOM), Mini-Conf.*, Shanghai, China, Apr. 2011, pp. 431–435.
[40] C. Sommer, D. Eckhoff, and F. Dressler, "IVC in cities: Signal attenuation by buildings and how parked cars can improve the situation," *IEEE Trans. Mobile Comput.*, vol. 13, no. 8, pp. 1733–1745, Aug. 2014.
[41] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen, "A survey on platoon-based vehicular cyber-physical systems," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 263–284, 3rd Quart., 2015.
[42] F. Klingler, F. Dressler, J. Cao, and C. Sommer, "MCB—A multi-channel beaconing protocol," *Ad Hoc Netw.*, vol. 36, no. 1, pp. 258–269, Jan. 2016.
[43] O. Papapetrou, W. Siberski, and W. Nejdl, "Cardinality estimation and dynamic length adaptation for Bloom filters," *Distrib. Parallel Databases*, vol. 28, nos. 2–3, pp. 119–156, Dec. 2010.
[44] K. Lee, U. Lee, and M. Gerla, "Geo-opportunistic routing for vehicular networks," *IEEE Commun. Mag.*, vol. 48, no. 5, pp. 164–170, May 2010.
[45] B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms* (Algorithms and Combinatorics), 5th ed. Springer, 2012, ch. 21.
[46] S. J. Swamidass and P. Baldi, "Mathematical correction for fingerprint similarity measures to improve chemical retrieval," *J. Chem. Inf. Model.*, vol. 47, no. 3, pp. 952–964, 2007.
[47] R. Michoud, A. Orozco, and G. Llano, "Mobile ad-hoc routing protocols survey for the design of VANET applications," in *Proc. IEEE Colombian Intell. Transp. Syst. Symp. (CITSS)*, Aug. 2012, pp. 1–6.
[48] J. Petit, F. Schaub, M. Feiri, and F. Kargl, "Pseudonym schemes in vehicular networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 228–255, Mar. 2015.
[49] *Intelligent Transport Systems (ITS); Security; Threat, Vulnerability and Risk Analysis (TVRA)*," document TS 102 893 V1.1.1, European Telecommunications Standards Institute, Mar. 2010.
[50] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *IEEE Trans. Mobile Comput.*, vol. 10, no. 1, pp. 3–15, Jan. 2011.
[51] C. Sommer, D. Eckhoff, R. German, and F. Dressler, "A computationally inexpensive empirical model of IEEE 802.11p radio shadowing in urban environments," in *Proc. 8th IEEE/IFIP Conf. Wireless Demand Netw. Syst. Services (WONS)*, Bardonecchia, Italy, Jan. 2011, pp. 84–90.
[52] C. Sommer and F. Dressler, "Using the right two-ray model a measurement based evaluation of PHY models in VANETs," in *Proc. 17th ACM Int. Conf. Mobile Comput. Netw. (MobiCom)*, Las Vegas, NV, USA, Sep. 2011.
[53] C. Sommer, S. Joerer, and F. Dressler, "On the applicability of two-ray path loss models for vehicular network simulation," in *Proc. Veh. Netw. Conf. (VNC)*, Nov. 2012, pp. 64–69.
[54] D. Eckhoff, A. Brummer, and C. Sommer, "On the impact of antenna patterns on VANET simulation," in *Proc. 8th IEEE Veh. Netw. Conf. (VNC)*, Columbus, OH, USA, Dec. 2016, pp. 17–20.
[55] C. Campolo, C. Sommer, F. Dressler, and A. Molinaro, "On the impact of adjacent channel interference in multi-channel VANETs," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 2626–2632.
[56] J. Heinovski, F. Klingler, F. Dressler, and C. Sommer, "Performance Comparison of IEEE 802.11p and ARIB STD-T109," in *Proc. 8th IEEE Veh. Netw. Conf. (VNC)*, Columbus, OH, USA, Dec. 2016, pp. 1–8.
[57] G. Pei and T. R. Henderson, "Validation of OFDM error rate model in NS-3," Boeing Res. & Technol., Tech. Rep., 2010.
[58] L. E. Miller, "Validation of 802.11a/UWB coexistence simulation," Nat. Inst. Standards Technol. (NIST), Gaithersburg, MD, USA, Tech. Rep., Oct. 2003.
[59] O. K. Tonguz, N. Wisitpongphan, and F. Bai, "DV-CAST: A distributed vehicular broadcast protocol for vehicular ad hoc networks," *IEEE Wireless Commun.*, vol. 17, no. 2, pp. 47–57, Apr. 2010.

Falko Dressler, photograph and biography not available at the time of publication.

Florian Klingler, photograph and biography not available at the time of publication.

Christoph Sommer, photograph and biography not available at the time of publication.

Reuven Cohen, photograph and biography not available at the time of publication.