

IEEE 802.11p Unicast Considered Harmful

Florian Klingler, Falko Dressler and Christoph Sommer

Distributed Embedded Systems Group, Dept. of Computer Science, University of Paderborn, Germany
 {klingler,dressler,sommer}@ccs-labs.org

Abstract—We study the feasibility of IEEE 802.11p unicast communication in Vehicular Ad Hoc Networks (VANETs). In brief, we found unicast communication using MAC acknowledgement frames (ACKs) to be unsuitable for vehicular networks, because missing ACKs make protocol operations susceptible to pronounced head of line blocking effects. Worse, a transmit queue that is blocked by a missing ACK will delay messages for all protocols running on the same node that use the same access category. Other than in traditional networks, missing ACKs are especially prevalent in VANETs due to their high topology dynamics. Depending on the scenario, delays of messages could be shown to reach 200ms and beyond – above the tolerable range of many VANET applications. Our findings are based on analytical calculations, measurements on hardware, as well as computer simulations; we conducted simulations both at small scale, for a baseline validation, and at a macroscopic level, to gauge the impact on a more complex protocol.

I. INTRODUCTION AND MOTIVATION

Wireless LAN (WLAN) according to the IEEE 802.11 standard has been widely adopted as the base technology for establishing vehicular networks, be it in the U.S. DSRC/WAVE stack, the European ETSI ITS-G5 stack, or the Japanese ARIB T-109 [1]. Of these, both the U.S. and the European stack inherit not just the physical and LLC layers of IEEE 802.11, but also the MAC layer.

Traditionally, the IEEE 802.11 WLAN MAC layer is designed to operate in the context of a Basic Service Set (BSS), a set of mobile nodes that have synchronized to use a common set of parameters [2]. However, joining a BSS (either an ESS in infrastructure mode, or an IBSS in ad hoc mode) requires an involved procedure that has been deemed too time consuming for vehicular networks. Hence, the WLAN standard has been amended in IEEE 802.11p to allow operation in “outside the context of a BSS” (OCB) mode [3], which has been introduced first as the Wireless Access in Vehicular Environments (WAVE) mode [4]. Operating in this mode obviates the need to authenticate to other nodes as well as the need to scan for, join, or associate to a BSS. This makes IEEE 802.11 a salient basis for vehicular networks: embedded systems can integrate off-the-shelf WLAN network interface cards with little or no modifications and still achieve low latency communication, which is crucial for safety applications.

What IEEE 802.11p networks (and thus, by extension, IEEE DSRC/WAVE and ETSI ITS-G5) also retain from the WLAN MAC, however, is its error control mechanism. At its core, WLAN realizes a simple Automatic Repeat Request (ARQ) error control mechanism: by default, any individually addressed (that is, any unicast) frame is not removed from its transmit

queue after being transmitted, but remains there until after an acknowledgment (ACK) frame is received.

If no ACK is received for a pre-set duration, frame transmission is automatically repeated until successful or until a pre-set limit is reached, all of which can add up to substantial amount of time. This is a problem because the ARQ mechanism causes head of line blocking. Any transmit queue that is waiting for an ACK for a unicast frame is stalled. The queue will neither transmit frames addressed to other cars, nor any broadcast frames – e.g., in the form of Cooperative Awareness Messages (CAMs) or Basic Safety Messages (BSMs) – which are crucial to support safety applications.

The head of line blocking effect has been identified in the early days of WLANs [5] and proposals have been made to create an alternative MAC layer that monitors the individual wireless stations of a BSS, maintaining separate transmit queues, and deferring re-transmissions to *bad* stations until the estimated end of a (presumed) burst error. However, the key assumption of such proposals has always been that lost frames are due to collisions or burst errors in the channel. In the envisioned target scenario, Mobile Ad Hoc Networks (MANETs), this was a very reasonable assumption due to relatively static topologies – and, here, the impact of the effect was no worse than reducing the attainable throughput over the wireless channel. This has led to the effect being widely ignored in standardization.

In vehicular networks, however, the effect of head of line blocking can be disastrous. First, the effect is easy to trigger. All that is needed is to either provoke unicast data transmission to a node that is not there. Alternatively, a denial of service attack can be mounted that is completely passive: Simply not acknowledging a unicast transmission allows a receiver to block all outbound transmissions from one of the sender’s queues for a substantial amount of time. Which queue is blocked depends on the Access Category allocated to the frame, but with only four categories defined in WLAN [2], a large number of different applications will likely share a single queue – hence, a single blocked queue impacts a multitude of related applications (for example, all safety applications). It is also impossible for the sender to tell whether the receiver suffers from interference that, indeed, keeps it from replying with ACKs – or whether ACKs are selectively suppressed, making this kind of attack hard, if not impossible, to detect reliably.

Second, the impact of head of line blocking is long-lasting. In MANETs, collisions and burst errors on the channel are merely a temporary reason for missing ACKs; re-sending a frame yields a high chance of success. In vehicular networks,

their highly dynamic network topology means that, frequently, the destination node is simply no longer a neighbor and remains permanently unreachable, e.g., due to radio signal shadowing [6]. This causes the transmit queue to block until the maximum number of retries have been exceeded, wasting channel capacity, keeping other nodes from transmitting, and (even worse) keeping the same node from transmitting potentially safety-critical information.

Our main contributions can be summarized as follows:

- We present related work around unicast communication in Vehicular Ad Hoc Networks (VANETs) (Section II);
- we investigate its impact analytically, in experiments, and in computer simulations (Section III); and
- we study the macroscopic impact of head of line blocking in a large scale computer simulation (Section IV).

II. RELATED WORK

Typical applications used in VANETs range from safety, to traffic efficiency, and to comfort applications [7]. To perform information dissemination for these kinds of applications several communication patterns have been found to be beneficial in VANETs [8].

The most important communication pattern for safety messages is beaconing, where vehicles periodically broadcast small status updates about their current position, speed, and driving direction. Usually this information exchange is strictly unidirectional and does not require any kind of acknowledgements.

On the other hand when using comfort applications, e.g., Internet access, nodes often have to communicate with a dedicated node or gateway. The preferred way for such connection oriented communication is to use unicast routing over multiple hops [8], [9]. Indeed, several detailed surveys on unicast routing protocols for VANETs can be found in the literature: Li and Wang [10] give an overview about different routing strategies and names popular routing protocols according to their communication type. Bernsen and Manivannan [11] classify and characterize available unicast routing protocols for VANETs and provide a qualitative comparison among those. Sichitiu and Kihl [12] focus on the taxonomy of VANET applications and study the requirements from an underlying network. Moreover they outline the differences between MANETs and VANETs and categorize routing protocols according to their addressing scheme. Many of those routing protocols have been originally developed for MANETs, and part of them can be applied to VANETs as well.

The unicast communication principle is also used in the literature for geocasting and platooning applications [13], [14]. The main objective is to provide reliable communication using retransmissions performed in the MAC layer. However when IEEE 802.11 was designed years ago, the exponential back off strategy for unsuccessful unicast communication triggered by lost ACKs was designed to solve channel congestion problems. The node topology was assumed to be relatively static, thus the most common causes for lost acknowledgements were assumed to be hidden terminal situations and, more importantly, an overloaded channel.

We show that for VANETs this assumption is not enough anymore; indeed, unicast communication drastically lowers the performance of VANETs when unicast packets are sent to nodes that are out of range, as is commonly happening for many protocols.

III. IMPACT ON MICROSCOPIC SCALE

We investigate the impact of unicast in VANETs first analytically, then in experiments with off-the-shelf WLAN cards and specialized equipment designed for Field Operational Tests (FOTs) worldwide, and finally in computer simulations.

A. Analytical Evaluation

In the following, we focus on an OFDM PHY with 10 MHz bandwidth as specified in the current version of the IEEE 802.11 standard [2]. We further assume that the RTS threshold is set above the frame size, so that no RTS/CTS procedure is invoked, as well as (otherwise) empty queues and an idle channel.

The time to transmit data is calculated according to the PLME-TXTIME.confirm primitive described in [2, Section 18.4.3]. When transmitting a payload l of 2400 bit at 6 Mbit/s, this time $t_{\text{tx-2400}}$ can be calculated as

$$\begin{aligned} t_{\text{tx-2400}} &= T_{\text{preamble}} + T_{\text{signal}} + T_{\text{sym}} \times \left\lceil \frac{16 + l + 6}{N_{\text{DBPS}}} \right\rceil \\ &= 32 \mu\text{s} + 8 \mu\text{s} + 8 \mu\text{s} \times \left\lceil \frac{16 + 2400 + 6}{48} \right\rceil \\ &= 448 \mu\text{s}. \end{aligned} \quad (1)$$

Similarly, for $l = 112$ bit, the size of an ACK frame, we obtain $t_{\text{tx-112}} = 64 \mu\text{s}$. The frame exchange sequence for (acknowledged) unicast transmission of a frame is *send data*, *wait for a SIFS*, *send ACK*.

Thus, the lower bound for head of line blocking (the case of the first transmission being successful) can be calculated as

$$\begin{aligned} t_{\text{unicast}} &= t_{\text{tx-2400}} + t_{\text{SIFS}} + t_{\text{tx-112}} \\ &= 448 \mu\text{s} + 32 \mu\text{s} + 64 \mu\text{s} \\ &= 0.544 \text{ ms}. \end{aligned} \quad (2)$$

If we now focus on the case of a node trying to send an (acknowledged) unicast frame to a node that does not exist, we have to factor in the time spent for retries, each waiting for an ACK that does not arrive $t_{\text{ACK_wait}}$, as well as the time spent in backoff. According to [2, Section 9.3.2.8], $t_{\text{ACK_wait}}$ can be calculated as

$$\begin{aligned} t_{\text{ACK_wait}} &= t_{\text{SIFS}} + t_{\text{SLOT}} + t_{\text{rx_delay}} \\ &= 32 \mu\text{s} + 13 \mu\text{s} + 49 \mu\text{s} \\ &= 94 \mu\text{s}. \end{aligned} \quad (3)$$

Backoff times are set to n times t_{SLOT} , the number n being randomly drawn from a contention window CW , which is initially set to CW_{min} ; in the worst case, the maximum number is drawn each time. After each unsuccessful transmission (i.e., no ACK was received) CW is updated to $((CW + 1) \times 2) - 1$, up to CW_{max} . Only when the packet is finally deleted from the transmit queue, CW is reset to CW_{min} .



Figure 1. Devices used in the experiments: embedded systems running Linux 3.9, outfitted with a UNEX DCMA-86P2 miniPCI card using the ath5k driver, and Cohda Wireless MK5.

For the following, we assume the default values suggested by [2, Page 1623, 2425]: $CW_{\min} = 15$, $CW_{\max} = 1023$, and $\text{DOT11SHORTRETRYLIMIT} = 7$ retransmission attempts. This configuration of the CW has been found to be beneficial to protocol operation in VANETs [15].

Taken together, the upper bound for head of line blocking in the described constellation can be calculated as

$$\begin{aligned} t_{\text{block,upper}} &= 8 \times (t_{\text{tx-2400}} + t_{\text{ACK_wait}}) + t_{\text{SLOT}} \times \\ &\quad (15 + 31 + 63 + 127 + 255 + 511 + 1023) \quad (4) \\ &= 30\,661 \mu\text{s}. \end{aligned}$$

Thus, each unicast sent to a node that no longer exists (whether sent in error or provoked maliciously) blocks any transmissions from the same queue for up to approx. 31 ms.

B. Experimental Study

We confirmed both the presence and the analytically derived duration of the blocking effect in real world experiments.

As our first device, we investigated an embedded system running Linux 3.9 and outfitted with a UNEX DCMA-86P2 miniPCI card using the ath5k driver (our full measurement setup is depicted in Figure 1). This card has been used by researchers worldwide such as Teixeira et al. [16], Santa et al. [17], Reis et al. [18], and many participants of the 2011 Grand Cooperative Driving Challenge,¹ such as Geiger et al. [19].

We modified the Linux kernel to amend Radiotap headers with how long each frame was delayed in a transmit queue – from entering into the queue to being deleted. We then configured an independent virtual interface set to monitoring mode to record these statistics.

As a baseline, we ran three independent applications on the device as depicted in Figure 2; all three sent 2400 bit frames to saturate an otherwise clear channel to a second device. The applications were designed so that all three queued their frames simultaneously, then waited for transmissions to conclude.

We configured the physical layer according to IEEE 802.11p specifications, using a 10 MHz wide channel at 5.890 GHz, not using RTS/CTS, and transmitting at a rate of 6 Mbit/s. The MAC layer was configured to send packets using a TXOP value of 0 (one post-transmit backoff after every frame) and

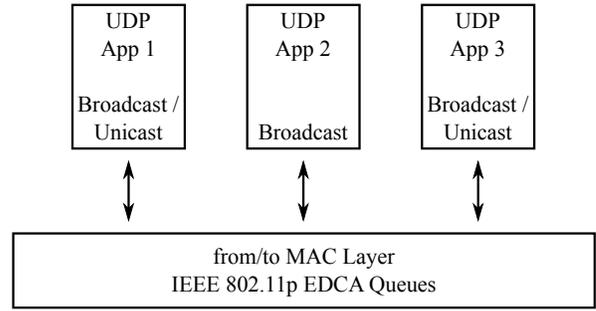


Figure 2. Experiment and Simulation Setup.

Access Category AC_BE, that is, an initial contention window size of 15 slots, a maximum contention window size of 1023 slots, and an AIFSN value of 2 slots, that is,

$$t_{\text{AIFS}} = t_{\text{SIFS}} + 2 \times t_{\text{SLOT}} = 58 \mu\text{s}. \quad (5)$$

In the first experiment (Exp 1), all three applications sent broadcast packets; in the second experiments application App 1 was changed to send unicast packets while App 2 and App 3 still sent broadcast packets.

Figure 3a illustrates the results: When sending only broadcast frames, no difference between App 1 and App 2 can be observed (as expected). Ignoring little delays introduced by the software, it can be seen that all data took either

$$t_0 = t_{\text{AIFS}} + t_{\text{tx-2400}} = 506 \mu\text{s} \quad (6)$$

to send (if no frame was already queued) or they had to wait for the frame of one or two of the other applications to be sent, corresponding to

$$t_1 = t_0 + \mathcal{U}(0, CW_{\min}) * t_{\text{SLOT}} + t_{\text{tx-2400}} \quad (7)$$

if one frame was queued, resulting in the interval 1012–1207 μs , as well as

$$t_2 = t_1 + \mathcal{U}(0, CW_{\min}) * t_{\text{SLOT}} + t_{\text{tx-2400}} \quad (8)$$

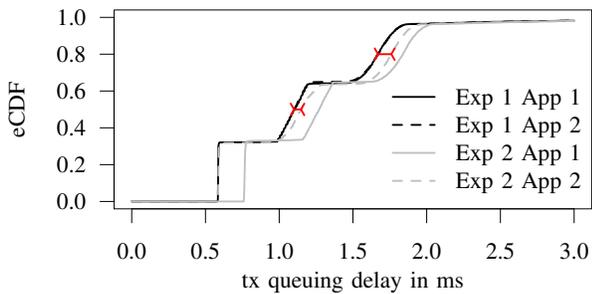
if two frames were already queued, resulting in the interval 1460–1850 μs .

Taken together, it can be said that experimental results are in perfect agreement with the analytical findings.

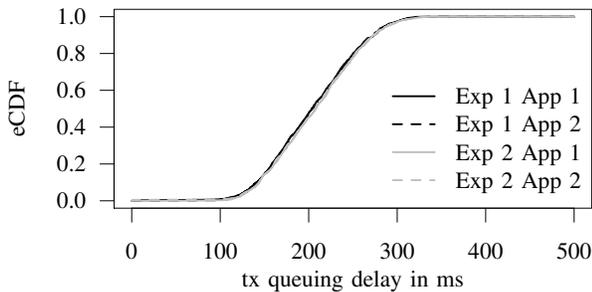
When changing App 1 to unicast (Exp 2), frames are delayed commensurate to the additional ACK frame that needs to be sent (and processed) – not just for App 1, which takes longer to send frames, but also for App 2 because of head of line blocking.

Figure 3b illustrates the results when App 3 was changed to transmit data to a device that was no longer there – thus representing the case of a vehicle trying to send data to a former neighbor – an effect that has been established to harm protocol performance in VANETs [20]. We manually inserted entries into the ARP tables of nodes in order to force the transmission even if no node is existent, thus, reproducing the scenario of a neighbor having existed previously before moving out of reception range. Also note that the queue size of the card's ath5k

¹<http://www.gcdc.net/>

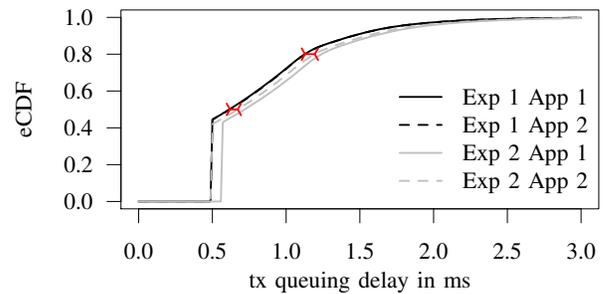


(a) App 3 sends broadcast frames.

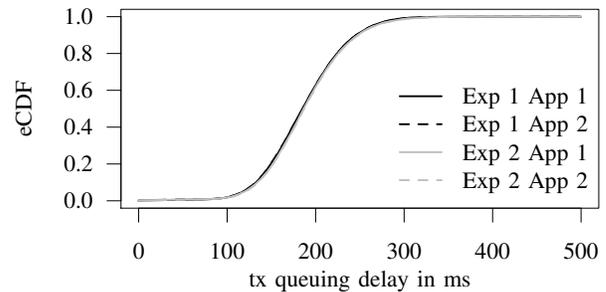


(b) App 3 sends unicast frames and loses ACKs.

Figure 3. TX queuing delay for two different experiments each (using UNEX DCMA-86P2 miniPCI cards).



(a) App 3 sends broadcast frames.



(b) App 3 sends unicast frames and loses ACKs.

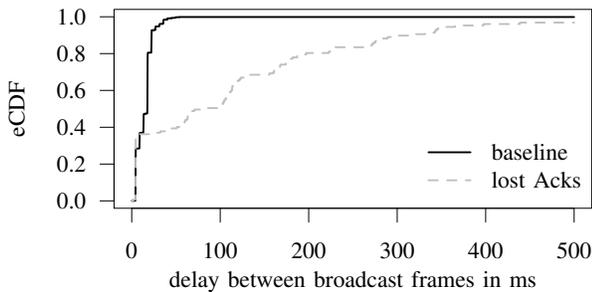
Figure 5. TX queuing delay for two different simulations each (using the *Veins* simulator).

Figure 4. Delay between broadcast frames for a baseline experiment with successful ACKs and one where ACKs are lost (using Cohda Wireless MK5).

driver was capped at 50 frames. It is immediately obvious that the lost ACKs of App 3 transmissions had a catastrophic effect on the delay of App 1 and App 2 transmissions (independent of whether these used unicast or broadcast).

Both applications' frames were queued for a typical duration of 207 ms and delay easily reached above 346 ms – well worse than the demands of many VANET applications [21], [22].

In order to confirm that this effect is not limited to commercial off-the-shelf WLAN devices, but also present in specialized equipment designed for FOTs worldwide, we ran the same test on a Cohda Wireless MK5, the company's newest model of integrated vehicular networking systems. Cohda Wireless devices have been used for major field trials like the simTD project in Germany and the U.S. Safety Pilot initiative [23]. Although we were not able to directly record the queuing time of frames for lack of access to the drivers, we were able to record the inter-frame interval. Figure 4 confirms

that the effect is just as pronounced here. This illustrates the grave effect that head of line blocking – provoked by unicast frames addressed to a former neighbor – has on broadcast frames' delay.

C. Computer Simulation

We validate our results on a microscopic scale by cross-checking the analytical and experimental results against a computer simulation of the same scenario, set up in the *Veins* Open Source² vehicular network simulation framework [24]. *Veins* provides realistic channel access models based on IEEE 802.11p and IEEE 1609.4 as well as realistic radio propagation models. We extended the IEEE 802.11p MAC layer in order to support unicast transmission according to the IEEE 802.11 HCF. As in the experiments, the MAC layer was configured to send packets using a TXOP value of 0 (one post-transmit backoff after every frame) and Access Category AC_BE, that is, an initial contention window size of 15 slots, a maximum contention window size of 1023 slots, and an AIFS value of 2 slots to match the settings used in the measurements.

In the baseline simulation, we let all three applications transmit frames with payload length of 2400 bit to saturate an otherwise clear channel to a second node.

Figure 5a shows the results for the baseline simulation. Unlike in the experiments, the simulated applications queued messages independent from each other, thus, the delays no longer fall into three clear categories according to how many (zero, one, or two) frames were queued before. Instead, it merely becomes less and less likely for incrementally larger

²<http://veins.car2x.org>

amounts of frames to queue before a given one (up to the configured maximum of 50 frames). Still, the lower bound of t_0 (for broadcast frames) and the additional delay for sending and processing ACKs (which is less than in the experiment, corresponding to no more than t_{SIFS} and $t_{\text{tx-112}}$) can very clearly be seen.

When sending only broadcast frames (Exp 1) around 40% of the data took no longer than t_0 to be transmitted. The remaining percentage of frames were queued behind other frames, waiting for them to be transmitted (or for the MAC to finish the post transmit backoff). When changing one application to transmit unicast packets (Exp 2), we see a similar effect like in the measurements. The additional delay of unicast frames introduced by the usage of acknowledgments also affects the broadcast transmissions of the other applications.

When we change the applications to transmit unicast traffic to a non existing address in the network as shown in Figure 5b, we see the same effect as in the measurements. Lost acknowledgements cause head of line blocking, increasing the delay frames spent in the transmit queue until they are removed.

Again, these results are in perfect agreement with the experimental results shown earlier, in Figure 3b.

IV. MACROSCOPIC VIEW

In order to investigate the impact of the discussed effects on a macroscopic scale, we conducted a computer simulation of a VANET. The VANET consisted of a large number of nodes running a typical protocol, which could be toggled between using broadcast or unicast communication. Again we employed the *Veins* Open Source vehicular network simulation framework, now making use of its coupling with the microscopic road traffic simulator SUMO.

We configured a freeway scenario with a length of 7 km in SUMO and performed network simulation in the center 5 km of the scenario. The 1 km border thus served to let the vehicles speed up and use realistic mobility patterns. We configured two different traffic densities on the freeway: 55 veh/km for a low utilized freeway, as well as 169 veh/km representing a high utilized freeway. Road traffic was modeled in SUMO sampling from a distribution of six different vehicle types (two types of trucks, and four types of cars) modeling different kinds of driving styles.

We collect results within a Region of Interest (ROI) of 3 km in order to not be influenced by border effects. The simulation warm-up period is configured to be 289 s to let the freeway get filled with vehicles, and another 11 s are used for the networking protocols to get into a steady state. Only after these 300 s we started to collect results. For all results, we plot the mean value together with the 95% confidence interval (please note that these intervals are sometimes very small). We repeated each simulation setup at least five times with different seeds for the random number generator in order to get statistically significant results.

A. Message Dissemination Protocol

To show the impact of unicast communication in VANETs we use a simple Geocasting protocol which disseminates information items among vehicles. Our Geocasting protocol maintains a knowledge base consisting of entries with geographic constraints and their expiration time.

In Figure 6, we show the building blocks of this protocol:

- **Neighbor Management:** Each vehicle broadcasts a beacon at a frequency of 1 Hz and maintains a 1-hop neighbor table (NT). Whenever vehicle v receives such a beacon from another vehicle u it adds u to the neighbor table \mathbb{N} . If two successive beacons are lost, in this case after 2 s, a node is removed from the neighbor table. This is performed right before information from the neighbor table is used.
- **Digest:** Whenever a vehicle v discovers a new neighbor u , it makes a probabilistic decision whether to inform this neighbor about information stored in the knowledge base (KB). With a probability of $p = \frac{1}{\text{new neighbors per s}}$ node u will be informed of the active events stored in the knowledge base of v . In this case v sends a small digest including fingerprints of all available events in the knowledge base, limited by the maximum frame size.
- **Data Request:** When node u receives a digest it responds with a data request including fingerprints of interesting information, called missing entries: An event is marked as missing, if the distance between u and the entry's destination position is lower than the distance between v and the entry's destination position, or if the vehicle is driving towards the destination direction. In other words, a node only selects an entry as missing, if it is closer to its destination position than the node which offers the entry, or the vehicle is driving to the destination.
- **Data Packet:** A node v which receives a data request from u constructs and sends a data packet to u containing all information which was marked as missing by u , again limited by the maximum frame size. This data packet can be overheard by all other neighbors using a monitor interface connected to the transceiver. When this data is received by any node w the knowledge base gets updated. If new information was contained in the data packet, w iterates over all neighbors in its neighbor table \mathbb{N} ; then, for each neighbor it takes a probabilistic decision with $p = \frac{1}{|\mathbb{N}|}$ whether to send a digest to node n .

Neighbor beacons use a different EDCA queue for transmission than the digest packets, data request packets, and data packets in order to not influence each other in terms of head of line blocking as outlined in Section III. In our simulations we have chosen AC_BE with an AIFS value of 2 slots for neighbor beacons, and AC_BK with an AIFS value of 9 slots for the rest. The CW_{\min} and CW_{\max} values for both EDCA queues are configured to be 15 and 1023 slots respectively. Moreover all packets except neighbor beacons can either be sent as broadcast, where the receiver address is annotated in the payload, or as unicast.

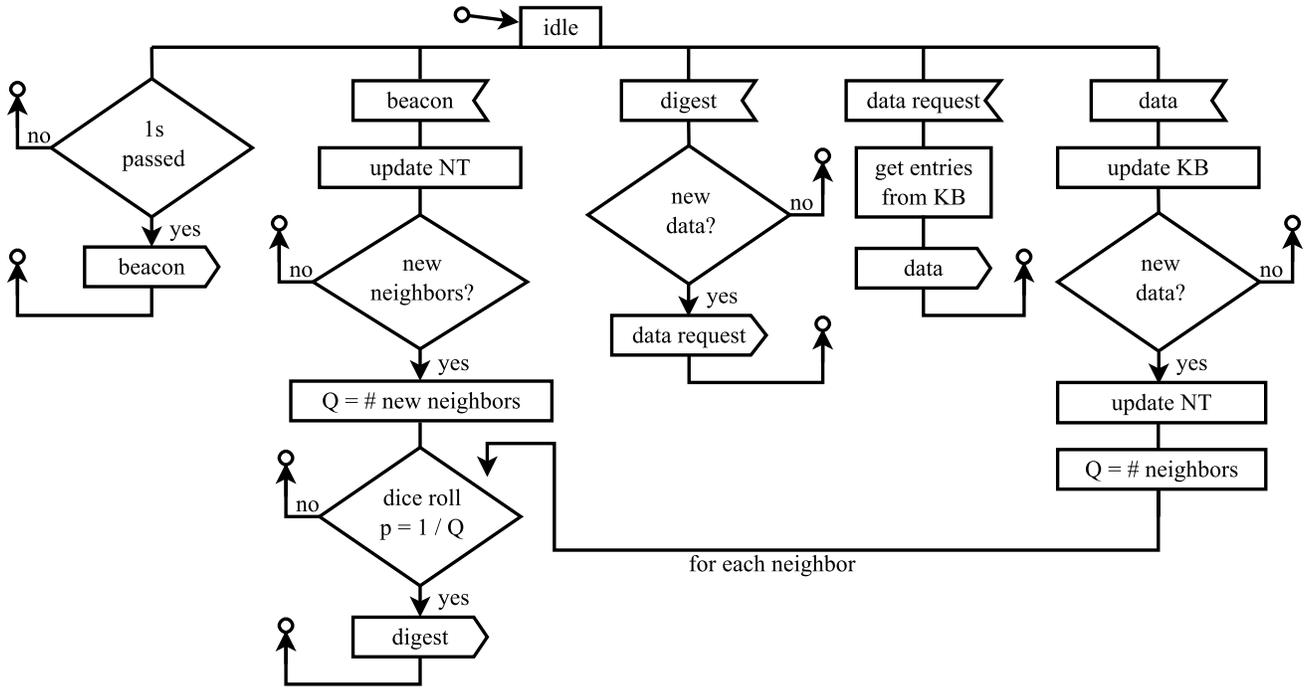


Figure 6. Message dissemination protocol.

B. Performance Metrics

An important metric to evaluate the neighbor table maintenance is the number of 1-hop neighbors for each node, as well as the correctness of this information. We compare the neighbor maintenance process against an oracle. This oracle calculates the neighbor information according to a unit disk model. For the distances of nodes to be treated as 1-hop neighbor we use the 99% quantile of 1-hop distances of our sample simulations for the communication distance. Thus, we are able to calculate the fraction of missing and outdated neighbors which represents the quality of the maintained neighbor information.

Finally we evaluate the drop rate of neighbors, meaning how many 1-hop neighbors were deleted from the neighbor tables of a vehicle due to lost beacons or because the node was not in range anymore. This gives an overview on the stability of neighbors tables.

For the Geocasting application the premier metric is the fraction of informed nodes for a specific information item. Besides that also the delay for receiving nodes to get this information plays an important role. In our simulation we generate new information items in the knowledge base of vehicles at each end of the ROI. The information items' destination position is at the opposite end of the ROI, meaning that each has to be disseminated through the whole network. After a simulation has reached a steady state, we create one information item, which we monitor while it traverses the network. We record the delay each node measures from generation of this information item until reception (and infinite delay otherwise). The simulation was configured to collect results for 15 s. We compare this Geocasting application for

two configurations: First, broadcast, meaning a node performs no retries and immediately goes into post transmit backoff after transmission of a frame. Second, unicast using MAC ACKs as defined by the IEEE 802.11 HCF and retransmissions if necessary. The maximum frame size was configured to be 1024 B, an information item in the knowledge base takes 64 B, and a digest takes 8 B per entry.

C. Evaluation

We performed two simulation studies: First, only looking at the performance of neighbor table maintenance. Second, including Geocasting for message dissemination.

1) *Neighbor Management*: In our simulation we observe a mean value of around 42 and 154 neighbors for each vehicle for the low and high density scenario respectively. Unfortunately the amount of neighbors is no indicator how accurate this information is. We therefore investigate the rate of outdated and missing neighbors compared to an oracle and measure around 4% outdated and 5% missing information for the

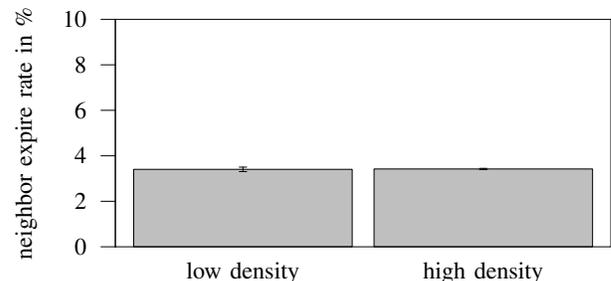
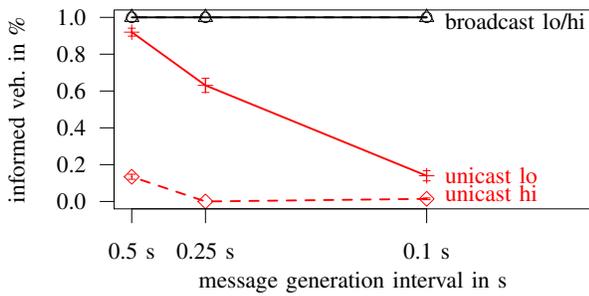
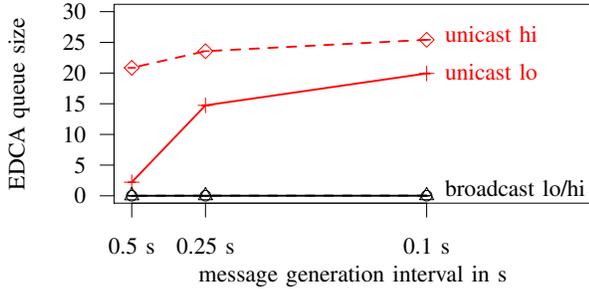


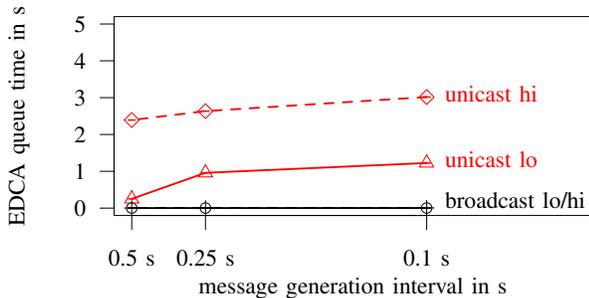
Figure 7. Neighbor table performance for different traffic densities.



(a) Fraction of informed vehicles.



(b) EDCA queuing size.



(c) EDCA queuing delay.

Figure 8. Performance of the Geocasting app for different traffic densities and message generation intervals.

low density scenario. In the high density scenario we measure around 3 % outdated and 10 % missing information respectively. To measure the stability of the neighbor tables we calculate the mean neighbor expire rate per second (that is, the churn rate of neighbors) shown in Figure 7. We note that the value remains constant for both traffic densities, indicating that the neighbor management process does not cause channel congestion.

Taken together, all results indicate that the churn rate of neighbors is very high: in addition to a number of wrong and missing entries, 3 % of entries need to be invalidated after each beacon interval.

2) *Geocasting*: In Figure 8a, we show the fraction of vehicles that received a particular information item for different traffic densities and message generation intervals. Note that both broadcast and unicast allow overhearing of information, thus a node can overhear unicast packets not designated to it (alike to running an additional interface in monitoring mode), handing their information up to the application layer. We compare these two schemes against each other.

Intuitively, we would expect a higher rate of informed nodes for unicast: after all, retransmissions would add reliability to wireless communication. As we can see the exact opposite is the case: Enforcing MAC ACKs and thus causing head of line blocking when ACKs are not received greatly reduces the performance of the VANET compared to broadcast communication.

To make it more clear, we want to highlight the two cases for lost ACKs:

- 1) Because of interference: when channel utilization is high, the chance of packet collisions increases. This can cause lost ACKs, either because a node did not receive the frame to respond with an ACK, or the ACK itself got lost. Both cases refer to an overloaded channel, in which exponential backoff and transmission retries makes sense and helps to reduce channel congestion and deliver the frame. From a brief look at channel load, however, we found interference to not be the predominant reason for missing ACKs in our VANET scenario. Instead, the reason lies in wrong entries of the neighbor table:
- 2) Because of neighbors that are gone: this case can especially happen in VANETs when receiving nodes are at the fringe of the communication distance of a sender, or when the communication channel is not symmetric – caused by obstacles, mobility or interference. When sending unicast frames enforcing MAC ACKs, and the receiver is not within networking distance anymore, exponential backoff and transmission retries to the same node make no sense. Moreover, exponential backoff has a negative impact in this case, since other applications suffer from a lower networking performance caused by head of line blocking.

When considering the accuracy of neighbor tables, we only observe a very small portion of outdated neighbors. Still, this low amount of unreachable neighbors hurt unicast communication such that head of line blocking occurs, and subsequent frames get delayed as explained in detail in Section III. On the other hand, broadcast based communication always reaches nearly 100 % of the nodes.

When looking at delays we see a similar picture (data not shown). The delay of unicast communication mode is always higher than with broadcasting.

Next, we consider the EDCA queue fill level. In Figure 8b, we show the average number of queued packets at the EDCA queue assigned for geocasting traffic. When using broadcast communication, the EDCA queue stays empty. For the unicast mode, queued elements add up due to head of line blocking caused by transmission attempts to nodes with no stable wireless link. Moreover, this effect can also be observed when looking at the delay of packets spent in the EDCA queue as seen in Figure 8c. The higher the number of queued elements, the longer each packet needs to be transmitted, thus increasing the end-to-end delay.

V. LESSONS LEARNED AND FUTURE WORK

In this paper, we studied the feasibility of unicast communication in VANETs for reliable message dissemination. Our findings are based on analytical calculations, measurements on hardware, as well as computer simulations.

In summary, we can say that unicast communication using MAC ACKs as defined per IEEE 802.11 HCF is unsuitable for vehicular networks to provide reliable communication. With experiments using UNEX DCMA-86P2 wireless cards and Cohda Wireless MK5 devices, we identified the head of line blocking problem when using unicast communication to unavailable nodes in a network – which is quite common in VANETs.

The main cause of nodes not acknowledging received packets in the network is rooted in their mobility, which frequently and quickly changes the network topology within a short period of time. When IEEE 802.11 was designed years ago, it was assumed that nodes stay at static positions or are only slowly moving. Thus, when performing unicast communication requiring MAC ACKs – and those ACKs are not received – a sending node performs exponential back-off and retransmissions. This is beneficial for static wireless networks, because lost ACKs are mainly caused by overloaded channels – either because a node did not receive the data to send an acknowledgement, or the ACK collided in a hidden terminal scenario.

In vehicular networks, this behavior is disastrous due to the high mobility of nodes. It can easily happen that a node is at the fringe of another node's communication distance, and further increases its distance to it. Thus, retransmissions and exponential back-off are useless, since frame reception probability decreases when the distance between nodes increases. Moreover, this behavior blocks subsequent frames to be transmitted (the so called head of line blocking) leading to a degraded performance of the network.

In order to provide reliable communication for VANETs, we cannot rely on acknowledged unicast anymore, as some protocols described in Section II do. One way forward could be to move the retransmission decision into the application layer in order to be able to make ARQ decisions based on multiple parameters. Moreover, it is necessary to take advantage of cross layer design and optimization utilizing application layer metrics. This will be part of the future work in the context of reliable communication using IEEE 802.11p.

REFERENCES

- [1] C. Sommer and F. Dressler, *Vehicular Networking*. Cambridge University Press, Nov. 2014, p. 384.
- [2] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE, Std 802.11-2012, 2012.
- [3] "Wireless Access in Vehicular Environments," IEEE, Std 802.11p-2010, Jul. 2010.
- [4] R. A. Uzcátegui and G. Acosta-Marum, "WAVE: A Tutorial," *IEEE Communications Magazine*, vol. 47, no. 5, pp. 126–133, May 2009.
- [5] P. Bhagwat, P. Bhattacharya, A. Krishna, and S. Tripathi, "Enhancing throughput over wireless LANs using channel state dependent packet scheduling," in *15th IEEE Conference on Computer Communications (INFOCOM 1996)*, San Francisco, CA: IEEE, Mar. 1996, pp. 1133–1140.
- [6] C. Sommer, S. Joerer, M. Segata, O. K. Tonguz, R. Lo Cigno, and F. Dressler, "How Shadowing Hurts Vehicular Communications and How Dynamic Beaconing Can Help," *IEEE Transactions on Mobile Computing*, vol. 14, no. 7, pp. 1411–1421, Jul. 2015.
- [7] F. Dressler, H. Hartenstein, O. Altintas, and O. K. Tonguz, "Inter-Vehicle Communication - Quo Vadis," *IEEE Communications Magazine*, vol. 52, no. 6, pp. 170–177, Jun. 2014.
- [8] E. Schoch, F. Kargl, M. Weber, and T. Leinmüller, "Communication Patterns in VANETs," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 119–125, Nov. 2008.
- [9] S. Yousefi, M. S. Mousavi, and M. Fathy, "Vehicular Ad Hoc Networks (VANETs): Challenges and Perspectives," in *6th International Conference on ITS Telecommunications (ITST 2006)*, Chengdu, China, Jun. 2006, pp. 761–766.
- [10] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: A survey," *IEEE Vehicular Technology Magazine*, vol. 2, no. 2, pp. 12–22, 2007.
- [11] J. Bernsen and D. Manivannan, "Unicast routing protocols for vehicular ad hoc networks: A critical comparison and classification," *Pervasive and Mobile Computing*, vol. 5, no. 1, pp. 1–18, Feb. 2009.
- [12] M. L. Sichertiu and M. Kihl, "Inter-Vehicle Communication Systems: A Survey," *IEEE Communications Surveys and Tutorials*, vol. 10, no. 2, pp. 88–105, 2008.
- [13] M. Kihl, M. Sichertiu, T. Ekeroth, and M. Rozenberg, "Reliable Geographical Multicast Routing in Vehicular Ad-Hoc Networks," in *Wired/Wireless Internet Communications*, Springer, 2007, pp. 315–325.
- [14] A. Böhm, M. Jonsson, K. Kunert, and A. Vinel, "Context-Aware Retransmission Scheme for Increased Reliability in Platooning Applications," in *Communication Technologies for Vehicles*, Springer, 2014, pp. 30–42.
- [15] R. Reinders, M. Van Eenennaam, G. Karagiannis, and G. Heijenk, "Contention window analysis for beaconing in VANETs," in *7th International Wireless Communications and Mobile Computing Conference (IWCMC 2011)*, Istanbul, Turkey: IEEE, Jul. 2011, pp. 1481–1487.
- [16] F. A. Teixeira, V. F. e Silva, J. L. Leoni, D. F. Macedo, and J. M. Nogueira, "Vehicular networks using the IEEE 802.11p standard: An experimental analysis," *Vehicular Communications*, vol. 1, no. 2, pp. 91–96, Apr. 2014.
- [17] J. Santa, F. Pereñíguez, A. Moragón, and A. F. Skarmeta, "Experimental evaluation of CAM and DENM messaging services in vehicular communications," *Transportation Research Part C: Emerging Technologies*, vol. 46, pp. 98–120, Sep. 2014.
- [18] A. B. Reis, S. Sargento, F. Neves, and O. K. Tonguz, "Deploying Roadside Units in Sparse Vehicular Networks: What Really Works and What Does Not," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 6, pp. 2794–2806, Jul. 2014.
- [19] A. Geiger, M. Lauer, F. Moosmann, B. Ranft, H. Rapp, C. Stiller, and J. Ziegler, "Team AnnieWAY's entry to the 2011 Grand Cooperative Driving challenge," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1008–1017, Apr. 2012.
- [20] W. Alasmay and W. Zhuang, "Mobility impact in IEEE 802.11p infrastructureless vehicular networks," *Ad Hoc Networks*, vol. 10, no. 2, pp. 222–230, Mar. 2012.
- [21] M. Segata, B. Bloessl, S. Joerer, C. Sommer, M. Gerla, R. Lo Cigno, and F. Dressler, "Towards Inter-Vehicle Communication Strategies for Platooning Support," in *7th IFIP/IEEE International Workshop on Communication Technologies for Vehicles (Nets4Cars 2014-Fall)*, Saint-Petersburg, Russia: IEEE, Oct. 2014, pp. 1–6.
- [22] T. K. Mak, K. P. Laberteaux, and R. Sengupta, "A multi-channel VANET providing concurrent safety and commercial services," in *2nd ACM International Workshop on Vehicular Ad Hoc Networks (VANET 2005)*, Cologne, Germany: ACM, Sep. 2005, pp. 1–9.
- [23] H. Rakouth, P. Alexander, A. Brown Jr., W. Kosiak, M. Fukushima, L. Ghosh, C. Hedges, H. Kong, S. Kopetzki, R. Siripurapu, and J. Shen, "V2X Communication Technology: Field Experience and Comparative Analysis," in *FISITA World Automotive Congress*, Beijing, China: Springer, Nov. 2012, pp. 113–129.
- [24] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, Jan. 2011.