<http://www.diva-portal.org>

Preprint

This is the submitted version of a paper presented at *5th International Workshop on Communication Technologies for Vehicles, Nets4Cars/Nets4Trains 2013, Villeneuve d'Ascq, France, May 14-15, 2013*.

Citation for the original published paper:

Jonsson, M., Kunert, K., Böhm, A. (2013)

Increased communication reliability for delay-sensitive platooning applications on top of IEEE 802.11p.

In: Marion Berbineau, Magnus Jonsson, Jean-Marie Bonnin, Soumaya Cherkaoui, Marina Aguado, Cristina Rico-Garcia, Hassan Ghannoum, Rashid Mehmood, Alexey Vinel (ed.), *Communication Technologies for Vehicles: 5th International Workshop, Nets4Cars/Nets4Trains 2013, Villeneuve d'Ascq, France, May 14-15, 2013. Proceedings* (pp. 121-135). Heidelberg: Springer Berlin/Heidelberg

Lecture Notes in Computer Science

http://dx.doi.org/10.1007/978-3-642-37974-1_10

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:hh:diva-24282>

Jonsson, M., K. Kunert, and A. Böhm, "Increased communication reliability for delay-sensitive platooning applications on top of IEEE 802.11p," M. Berbineau et al. (Eds.): Nets4Cars/Nets4Trains 2013, LNCS 7865, Springer-Verlag Berlin Heidelberg, pp. 121-135, 2013.

Increased Communication Reliability for Delay-Sensitive Platooning Applications on top of IEEE 802.11p

Magnus Jonsson, Kristina Kunert, Annette Böhm

CERES – Centre of Research on Embedded Systems, Halmstad University, Halmstad, Sweden
{magnus.jonsson,kristina.kunert,annette.bohm}@hh.se

Abstract. Cooperative driving in platooning applications has received much attention lately due to its potential to lower fuel consumption and improve safety and efficiency on our roads. However, the recently adopted standard for vehicular communication, IEEE 802.11p, fails to support the level of reliability and real-time properties required by highly safety-critical applications. In this paper, we propose a communication and real-time analysis framework over a dedicated frequency channel for platoon applications and show that our retransmission scheme is able to decrease the message error rate of control data exchange within a platoon of moderate size by several orders of magnitude while still guaranteeing that all delay bounds are met. Even for long platoons with up to seventeen members the message error rate is significantly reduced by retransmitting erroneous packets without jeopardizing the timely delivery of regular data traffic.

Keywords: Platooning, real-time communication, vehicular communication, retransmission scheme, real-time scheduling analysis.

1 Introduction

Intelligent Transport Systems (ITS) in general and cooperative driving in particular have received much attention from both researchers and media recently. Enabled by sensing and communication capabilities, as well as recent international standardization efforts, the introduction of cooperative driving shows great potential for increased safety and efficiency on our roads. The basic building blocks of a variety of ITS application are the exchange of status information [1] within a concerned group of vehicles, making it possible for the driver or the system itself to e.g. adjust to sudden changes in the current traffic situation. These changes usually need to be performed within a strict deadline and with high demands on reliability.

An application area where real-time and reliability demands are particularly obvious is platooning. It has been shown that a considerable reduction in fuel consumption can be achieved for trucks driving in platoons with reduced vehicle-to-vehicle spacing [2]. Additionally, the controlled and thereby more predictable behaviour of trucks on a highway increases the overall road safety not only for platoon members but for all surrounding road users. To enable a distance between platoon vehicles of 10 meters or

adfa, p. 1, 2011.

© Springer-Verlag Berlin Heidelberg 2011

The final publication is available at link.springer.com/chapter/10.1007/978-3-642-37974-1_10

less control data has to be exchanged within the platoon at a regular basis and with a high and predictable success probability.

The recently approved IEEE 802.11p standard for inter-vehicle communication [3] defines two message types, periodic status updates and event-triggered warning messages, that are allowed to share the dedicated ITS control channel in the 5.9 GHz ITS frequency band. To access that shared communication resource the standard employs CSMA/CA, a random access Medium Access Control (MAC) method, which by definition does not support the reliable and timely exchange of safety-critical and highly deadline-dependent control data needed for platoon control applications. Furthermore, there is no support for retransmissions to tackle the potentially high message error rate found in highly mobile wireless communication networks. Measurements of the packet reception ratio for different scenarios using IEEE 802.11p are reported in [4].

In this paper we therefore propose two enhancements used on top of the unaltered IEEE 802.11p standard in form of real-time functionality containing a deterministic, polling-based MAC approach as well as a transport layer retransmission scheme over a specific service channel dedicated to inter-platoon communication only. Both concepts are explained in detail in Chapter 2 and 3 in the context of a platooning scenario. Our approach improves the packet error rate (PER) in the platooning use case by several orders of magnitude while still maintaining a reasonable platoon size and is based on earlier work on infrastructure supported communication [5]. The inadequacy of IEEE 802.11p to obtain the level of reliability required in platoon control applications was recognized by [6] where infrared is suggested as a complementary communication technology for increased reliability. This adds additional hardware costs that our IEEE 802.11p-based approach manages to avoid. A polling-based MAC scheme similar to ours was presented in [7]. The lack of a real-time schedulability test, however, fails to provide timing guarantees. The authors of [8] show how to obtain strict priorities in 802.11p, but collisions among equal-priority packets can still appear. STDMA has been suggested as a promising alternative to the 802.11p MAC method [9], but even though the medium access delay is bounded, there still remains a possibility of collision. Studies on how to increase quality of service through retransmissions have been presented before. The combination of retransmissions with deadline-dependent traffic, however, is studied less thoroughly. [10] presents a retransmission scheme where packets only are retransmitted in case their deadline has not already passed, but the lack of any queuing analysis might imply the fact that the worst case delay of packets cannot be upper-bounded. [11] provides guarantees for timely treatment of real-time traffic, but lacks timing details and uses an analysis method shown to result in lower network utilization as the analysis method used in our approach [12]. For more general information and surveys on vehicular communication, see, e.g., [13-15].

2 Protocol Details

Imagine a platoon of trucks driving at reduced inter-vehicle distance at relatively high speed. To maintain that distance, every platoon member has to be well informed of the behaviour of its surrounding platoon neighbours, especially in case sudden changes in speed call for immediate actions performed by the vehicles' control systems. It can be assumed that one particular vehicle within the platoon assumes a central role in collecting the necessary status information, processes it and spreads the necessary

control data to the platoon members. This role often falls to the first vehicle in the platoon, the platoon leader. Furthermore, our proposed MAC method is based on a polling scheme administered by one node in the system in a master-slave fashion. As the length of a platoon easily can exceed the transmission radius of the platoon, we assign that role to a vehicle in the middle of the platoon, ensuring the best platoon coverage.

As the ITS control channel only allows the aforementioned IEEE 802.11p message to be transmitted, the polling messages our MAC approach relies on are not accepted on that channel. Due to the frequent and highly deadline-dependent data traffic needed to make a platoon function properly and safely it is reasonable to expect that a second transceiver will be available in platoon members, dedicated to platoon safety-related data exchange. Besides the ITS control channel there are several service channels available that can be used for that purpose. By using a service channel, we are not restricted to a specific message type and, at the same time, we are able to use the entire bandwidth of that service channel for platooning purposes while still listening to and participating in the data exchange with other road traffic users over the ITS control channel.

2.1 Medium Access Control

In the MAC protocol defined for e.g. IEEE 802.11a/b/g used for Wireless Local Area Networks (WLAN) the bandwidth is divided into Contention-Based Phases (CBPs) and Collision-Free Phases (CFP). The former is based on CSMA/CA random access where nodes compete for channel access and packet collisions are possible and, in the case of heavy channel utilization, even likely. The latter uses a polling-based scheme granting a specific node undivided access to the channel for a given period of time. As a slimmed version of the original IEEE 802.11 protocol suite, 802.11p does not offer collision-free channel access, despite the fact that many of its applications are safety-critical and put very high requirements on the timeliness and reliability. As proposed in previous work [16] [17], we reintroduce the possibility to use the CFP and propose a polling scheme administered by the master vehicle by placing a real-time layer on top of the IEEE 802.11p MAC layer. Admission control comprises a real-time schedulability test that insures the timely treatment of safety-critical data.

Time is divided into superframes (see Fig. 1) consisting of a CFP and a CBP, including a beacon frame, where channel access during the CBP is simply handled by IEEE 802.11p MAC in its random access fashion. Depending on the types of data traffic present in the network, the CBP can either be exploited by non-real-time packets in case those are present or reduced to zero (except for the beacon frame), freeing all bandwidth for safety-critical real-time traffic handled during the CFP. The beacon is used for synchronization purpose, but can also be used for, e.g., special short-message services [18] coordinated by the master. The transport layer hands down packets in Earliest Deadline First (EDF) [19] order and only at a point in time when the polling mechanism can ensure that the medium is available, thereby ensuring a deterministic treatment of the packet.

The master utilizes a polling scheme to grant the other vehicles access to the medium during the CFP. Knowing real-time traffic demands in the network, the polling

scheme schedules traffic according to EDF. In case data traffic is to be sent from a vehicle to the master, the master will poll the vehicle and allow it exclusive access to the medium. In case the data traffic flow is from the master to any other vehicle, the master will allocate time for its own transmission, including an ACK answer from the receiving vehicle. The polling protocol enables deterministic medium access during the CFP for real-time communication necessary for the platooning application.

Fig. 1. Superframe structure

2.2 Real-Time Transport Layer

In order to make it possible to decide upon retransmissions on a message level, the transport layer added to the protocol stack implements EDF scheduling of both ordinary transmissions and retransmissions, and a real-time ARQ (Automatic Repeat Request) scheme. The ARQ scheme allows us to retransmit erroneous packets before their deadline expires and without negative interference with other real-time data traffic. This scheme has been proven successful by the authors in other types of networks (industrial networks [20] [21] and infrastructure-based vehicular networks [5]) and is here applied to the specific requirements of a platooning scenario.

Real-time data traffic in the network is modelled as periodic real-time channels (RTCs) called τ_i (with $i = 1, 2, \dots$) and defined as $\tau_i = \{S_i, R_i, P_i, L_i, D_i\}$, where S_i is the sending node, R_i the receiving node, P_i the period, L_i the packet length, and D_i the relative transport layer deadline of the traffic flow. In order to have time for retransmissions before the end of the deadline, D_i is divided into two components: the deadline for the ordinary transmission ($D_{ord,i}$), and the deadline for retransmission(s) ($D_{retr,i}$). Retransmissions are modelled as retransmission real-time channels (ReRTCs) and defined by $\tau_{re,i} = \{P_{re,i}, L_{re,i}, D_{re,i}\}$. The period is a predetermined system parameter indicating the minimum interval possible between two retransmissions on the ReRTC and by that upper-bounding the bandwidth dedicated to retransmissions. The length specified is the maximum packet length in order to accommodate any retransmission packet, and the deadline is the maximum time allowed for the current retransmission.

Retransmission of a data packet from the master is triggered if no ACK is received directly following the data packet, i.e., before time out. In case the master does not receive a packet it polled for, it will simply schedule a retransmission of its polling packet (no link layer acknowledgements or retransmissions from the underlying standard are used). This mechanism leads to that merely the master has to make the decisions about retransmissions. Regular vehicular nodes only need packet queues for

ordinary transmission and a possibility to store packets in case they need to be re-transmitted. However, all packets can be discarded after their deadline. ACKs are sent directly upon the reception of a correct data packet (after the predefined highest-priority arbitration interframe spacing (AIFS)). Erroneous packets are detected at the MAC layer and not handed to the transport layer at all.

The master vehicle will schedule all ordinary transmissions and retransmissions in the form of EDF queues of polling and data packets. No packet can be sent before knowing how long it would take to transmit (T_{wait}), as no transmission can be initiated if T_{wait} is longer than the time remaining in the current CFP. These times are different for polling ($T_{wait,poll,i}$) and data packets ($T_{wait,data,i}$) and are calculated as follows:

$$T_{wait_poll,i} = T_{procM} + T_{AIFS} + T_{poll} + T_{prop} + T_{procS} + T_{AIFS} + T_{data,i} + T_{prop} + T_{procM_CRC} + T_{margin} \quad (1)$$

$$T_{wait_data,i} = T_{procM} + T_{AIFS} + T_{data,i} + T_{prop} + T_{procS_CRC} + T_{AIFS} + T_{ACK} + T_{prop} + T_{procM} + T_{margin} \quad (2)$$

where

- T_{procM} = processing time at the master before the start of the packet transmission
- T_{AIFS} = Highest-priority arbitration interframe spacing (always needed)
- $T_{poll} = L_{poll} / r$ = transmission time of a polling packet, where L_{poll} denotes the length of a polling packet and r the bit rate
- T_{prop} = maximum propagation delay
- T_{procS} = processing time at the slave after reception of packet
- $T_{data,i} = L_i / r$ = transmission delay of a data packet of RTC i
- T_{procM_CRC} = processing time at the master for error checking in data packet
- T_{margin} = time margin compensating for, e.g., other protocol delays
- T_{procS_CRC} = processing time at the slave for error checking in data packet
- $T_{ACK} = L_{ACK} / r$ = transmission time of ACK, L_{ACK} being the ACK packet length

Having sent a data or polling packet, the master waits the calculated waiting time before moving on to the next packet in the EDF queue. However, at the end of the waiting time, the master's transport layer will first check for a data or ACK packet handed up from the MAC layer. In case the received data packet was erroneous or no ACK arrived, the possibility of retransmissions has to be investigated. A packet can only be retransmitted a predefined number of times ($N_{attempt}$). Moreover, a ReRTC has to be available, i.e., it must not have been used during the length of one retransmission period P_{re} . Only if both checks are positive, a retransmission can be scheduled in the EDF queue, otherwise the packet has to be discarded.

3 Real-Time Analysis

In the following, a timing and feasibility analysis is described, adapting the analysis framework introduced in [22] [23] to work with this platooning single-hop star topology network with a central master node as defined above.

3.1 Timing Parameters

We are assuming superframes consisting of one CBP of length T_{CBP} and one CFP of length T_{CFP} . Data traffic with real-time demands will only be sent during the CFP. Considering a worst-case situation, not the whole CFP can be utilized for transmissions, as the last interval left during this phase might be too short to schedule the next packet from the queue. The length of this interval is called blocking time and denoted T_b . In our worst case analysis, T_b corresponds to the time it would take to transmit the longest packet pertaining to any of the traffic flows in the network, i.e.,

$$T_b = \max \left\{ \max_i \{T_{wait_poll,i}\}, \max_i \{T_{wait_data,i}\} \right\}. \quad (3)$$

In consequence, the length of the reduced CFP, T_{rCFP} , is calculated as:

$$T_{rCFP} = T_{CFP} - T_b. \quad (4)$$

As real-time traffic can only be sent during the length of the reduced CFP, we model the bit rate for the real-time traffic as an experienced bit rate r_e , i.e., we scale down the actual bit rate r by the ratio of the reduced CFP and superframe length, i.e.,

$$r_e = \frac{T_{rCFP}}{T_{SF}} \cdot r. \quad (5)$$

This leads to a scaling down also of the waiting times defined in Eq. 1 and 2:

$$T_{e,wait_poll,i} = T_{wait_poll,i} \cdot \frac{T_{rCFP}}{T_{SF}}, \quad (6)$$

$$T_{e,wait_data,i} = T_{wait_data,i} \cdot \frac{T_{rCFP}}{T_{SF}}. \quad (7)$$

3.2 Timing Analysis

For the analysis we are assuming that the number of RTCs in the network is Q , while there are M ReRTCs available. In conjunction with admission control (if not done offline during design stage), the master uses the real-time schedulability analysis described in the next chapter to only admit data traffic for which required delays can be

guaranteed. One of the parameters needed in this analysis is the maximum queuing delay that any packet can experience, and which can be calculated knowing the transport layer deadline D_i and the worst case delay the packet can experience. As D_i is divided into $D_{ord,i}$ and $D_{retr,i}$, the worst case delay consists of two components, $d_{ord,i}$ and $d_{retr,i}$, respectively. For the highest priority packet this will occur if the packet arrives to the EDF queue just as the transmission of a lower priority packet starts and if after the completion of this transmission the interval left in the CFP is just too short for the delayed packet to be sent. The worst case delay experienced by that packet would in this case comprise two waiting times plus a complete CBP and consequently the maximum queuing delay can be calculated as:

$$d_{ordM2S} = D_{ord,i} - T_{CBP} - T_{wait_data,i} - \max\{T_{wait_data,i}, T_{wait_poll,i}\}, \quad (8)$$

$$d_{ordS2M} = D_{ord,i} - T_{CBP} - T_{wait_poll,i} - \max\{T_{wait_data,i}, T_{wait_poll,i}\}. \quad (9)$$

When calculating the maximum queuing delay for packets using retransmission channels, d_{retr} , we assume the retransmission deadline $D_{retr,i}$ to be the same for all values of i , i.e., we can set $D_{retr,i} = D_{retr}$. Additionally, D_{retr} has to be divided between all of the allowed retransmission attempts:

$$D_{re} = \frac{D_{retr}}{N_{attempt}}. \quad (10)$$

The maximum queuing delay for retransmission packets can then be calculated as:

$$d_{retrM2S} = D_{re} - T_{CBP} - T_{wait_data,i} - \max\{T_{wait_data,i}, T_{wait_poll,i}\}, \quad (11)$$

$$d_{retrS2M} = D_{re} - T_{CBP} - T_{wait_poll,i} - \max\{T_{wait_data,i}, T_{wait_poll,i}\}. \quad (12)$$

3.3 Real-Time Scheduling Analysis

In the admission control, the master uses a real-time schedulability analysis originally developed for EDF scheduling of periodic tasks on a uniprocessor [19] in order to check the feasibility of the real-time traffic allocation as proven possible in [24]. The check is implemented in two stages, a utilization check and a workload check, and is only done when a new RTC is to be allocated. In case both checks are positive, the deadline for the checked traffic flow can be guaranteed.

The utilization check calculates the bandwidth utilization U of all RTCs and ReRTCs during the CFP and ensures it to be no bigger than 1. The calculation is done as follows:

$$\begin{aligned}
U = & \sum_{i=1}^{\alpha} \left(\frac{T_{e,wait_poll,i}}{P_i} \right) + \sum_{i=1}^{\beta} \left(\frac{T_{e,wait_data,i}}{P_i} \right) + \\
& \sum_{i=1}^{\gamma} \left(\frac{T_{e,wait_retr_poll,i}}{P_{re,i}} \right) + \sum_{i=1}^{\delta} \left(\frac{T_{e,wait_retr_data,i}}{P_{re,i}} \right). \tag{13}
\end{aligned}$$

Greek lettering indicates the number of traffic flows in the different summations. While α is the numbers of RTC from the master to the slaves, β denotes RTCs in the opposite direction, and $\alpha + \beta = Q$. γ and δ are the number of ReRTCs from master to slave and slave to master, respectively, where $\gamma + \delta = M$. $T_{e,wait_retr_poll}$ and $T_{e,wait_retr_data}$ are defined in the same manner as $T_{e,wait_poll}$ and $T_{e,wait_data}$, and only introduced here to help understanding.

The second check calculates the workload that is put onto the network by all accepted RTCs. It is done by the means of a workload function, $h(t)$, in which all transmission times of all packets of all traffic flows (including the retransmission flows) with a deadline before time t are summed up. Time starts at the start of a hyperperiod, i.e., when all traffic flows' periods start simultaneously. Hyperperiods continue until the periods start at the same time again. Using the workload check during one hyperperiod has been shown to constitute the worst case scenario [25-27]. The $h(t)$ function is calculated as follows, where summands one and two calculate the workload imposed by RTCs and summands three and four that of the ReRTCs:

$$\begin{aligned}
h(t) = & \sum_{\substack{i \in [1, \alpha], \\ d_{ordS2M,i} \leq t}} \left(1 + \left\lfloor \frac{t - d_{ordS2M,i}}{P_i} \right\rfloor \right) \cdot T_{e,wait_poll,i} + \\
& \sum_{\substack{i \in [1, \beta], \\ d_{ordM2S,i} \leq t}} \left(1 + \left\lfloor \frac{t - d_{ordM2S,i}}{P_i} \right\rfloor \right) \cdot T_{e,wait_data,i} + \\
& \sum_{\substack{i \in [1, \gamma], \\ d_{retrS2M,i} \leq t}} \left(1 + \left\lfloor \frac{t - d_{retrS2M,i}}{P_{re,i}} \right\rfloor \right) \cdot T_{e,wait_retr_poll,i} + \\
& \sum_{\substack{i \in [1, \delta], \\ d_{retrM2S,i} \leq t}} \left(1 + \left\lfloor \frac{t - d_{retrM2S,i}}{P_{re,i}} \right\rfloor \right) \cdot T_{e,wait_retr_data,i} \tag{14}
\end{aligned}$$

For the workload check to hold, the following demand has to hold:

$$h(t) \leq t \quad \forall t. \tag{15}$$

According to [28], this computationally heavy computation (due to the complexity introduced by the notion of continuous time) can be simplified to only include the following points in time at which the workload function has to be evaluated:

$$\begin{aligned}
t \in & \bigcup_{i=1}^{\alpha} \{w \cdot P_i + d_{ordS2M,i} : w = 0, 1, 2, \dots\} + \\
& \bigcup_{i=1}^{\beta} \{x \cdot P_i + d_{ordM2S,i} : x = 0, 1, 2, \dots\} + \\
& \bigcup_{i=1}^{\gamma} \{y \cdot P_{re,i} + d_{retrS2M,i} : y = 0, 1, 2, \dots\} + \\
& \bigcup_{i=1}^{\delta} \{z \cdot P_{re,i} + d_{retrM2S,i} : z = 0, 1, 2, \dots\}
\end{aligned} \tag{16}$$

and

$$t \in [1; P_{busy}]. \tag{17}$$

The busyperiod P_{busy} is the length of time between the start of a hyperperiod until the the link becomes idle the first time. Here P_{busy} denotes the first busyperiod, i.e., the busyperiod that starts at the beginning of the first hyperperiod.

4 Performance Evaluation

We evaluated the performance of our framework by simulating a series of platooning test cases and present the results from two representative cases in this chapter. The simulator was implemented in MatLab. The bit rate is set to be 6 Mbps and the frame durations are calculated and set according to IEEE 802.11p. A polling packet has aduration of 154 μ s, corresponding to a packet length of 80 bits plus MAC and physical layer headers and highest priority AIFS. ACKs are assumed to have the same duration as polling packets, while beacon frames have a duration of 370 μ s, comprising 200 bytes including MAC and physical layer headers and highest priority AIFS. The corresponding values for data packets are 642 μ s and 400 byte. To account for a deterioration of the channel quality due to fading and shadowing (by vehicles situated in-between sender and receiver), the simulated packet error rate is calculated as:

$$PER = N_{hop} \cdot 0.05 \tag{18}$$

where $N_{hop} = 1$ if the sending and receiving vehicle are direct neighbors, $N_{hop} = 2$ if there is one vehicle in-between etc.

In this performance study, we compare our retransmission scheme to a case without retransmissions. For each simulation setup, the average performance is based on

250 simulations runs of a duration of 1000 hyperperiods. An initial real-time feasibility check determines whether the timing requirements of a given task set can be met. Each vehicle is assumed to have one RTC for transmission to the master (middle vehicle), while the master has one RTC to the platoon leader and a RTC has a period and deadline of 50 ms. Due to the broadcast nature of the transmissions, every vehicle can overhear and make use of all data traffic sent within its reception range. Only the single destination RTCs described in Chapter 3, however, are simulated and considered for evaluation.

In the first simulated case, Case 1, we have a superframe length of 25 ms and a CBP length of 5 ms (including beacon). There are $M = 4$ retransmission RT channels, each having a period of $P_{re,i} = 25$ ms. The maximum number of retransmission attempts is set to $N_{attempt} = 2$. The deadline for the retransmission RT channels is always calculated automatically to the minimum possible value by using the workload function and the real-time analysis backwards. With this configuration, the real-time analysis showed that a maximum of seventeen vehicles could be supported without deadline misses. Fig. 2 shows the experienced message error rate (MER) without ARQ, after maximum one or, if required, two retransmission attempt. As the number of vehicles (and RTCs) increases, the retransmission RTCs become a bottleneck and retransmission attempts must be rejected. Moreover, the PER increases with distance. Disregarding that fact in a simulation with a distance-independent PER (not shown here) we could still see a higher average MER in longer platoons. As seen in Fig. 2, a MER below 10^{-3} can be achieved with only a few vehicles in the platoon, while the MER reaches 6.3×10^{-2} for 17 vehicles, still a considerable reduction compared to the case without retransmissions. Fig. 3 shows the MER for the master only, reaching 0.13 for 17 vehicles and a reduction of one to two orders of magnitude for moderate-sized platoons. Fig. 4 shows that the channel busy time, i.e. the percentage of time when the medium is occupied, in most cases stays below 25% and that penalty introduced by retransmissions is moderate. The average delay (not shown) for all platoon lengths remains below 2.2 ms and thereby far below the stated guarantee.

In the second simulated case, Case 2, we have a superframe length of 50 ms and no CBP. There are $M = 9$ retransmission RTC, each with a period of $P_{re,i} = 20$ ms. The maximum number of retransmission attempts is set to $N_{attempt} = 3$. According to the real-time analysis, this configuration allows for a maximum of 15 vehicles in the platoon. Fig. 5 compares the MER without retransmissions and to cases of up to three retransmission attempts where a reduction by roughly two and three orders of magnitude can be achieved for longer and shorter platoons, respectively.

Fig. 2. Message error rate for Case 1

Fig. 3. Message error rate for master to leading vehicle in Case 1

Fig. 4. Channel busy time for Case 1

The MER for the master sending to the leading vehicle is shown in Fig. 6. For 15 vehicles the reduction in MER is more than one order of magnitude. For shorter platoons an improvement of several orders of magnitude can be achieved. The average delay (not shown) is below 1.3 ms for all platoon sizes, while the channel busy time (not shown) reaches 30 % for a platoon length of 15 vehicles and 25% for 13 vehicles, respectively.

5 Conclusion

In this paper we presented an enhancement to the IEEE 802.11p standard for the reliable and timely exchange of safety-critical control data in platooning applications. The framework combines guaranteed delay bounds with the possibility of retransmitting erroneous packets. Although we showed that our retransmission scheme significantly reduces the experienced message error rate, our simulation studies even shed light on the overall effect of the number of platoon members on the reliability that can be achieved. For moderate sized platoons, a reduction in message error rate by several orders of magnitude could be achieved.

Fig. 5. Message error rate for Case 2

Fig. 6. Message error rate for master to leading vehicle in Case 2

6 References

1. Vinel, A., Staehle, D., Turlikov, A.: Study of Beaconing for Car-to-Car Communication in Vehicular Ad-Hoc Networks. In: Proc. IEEE International Conference on Communications, Dresden, Germany, June. IEEE (2009)
2. Alam, A., Gattami, A., Johansson, K.H.: An Experimental Study on the Fuel Reduction Potential of Heavy Duty Vehicle Platooning. In: International Conference on Intelligent Transportation Systems, Madeira, Portugal, Sept.. IEEE (2010)
3. Standards Committee, IEEE Standard 802.11p-2010, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments", pp.1-51, July (2010)
4. Böhm, A., Lidström, K., Jonsson, M., Larsson, T.: Evaluating CALM M5-based Vehicle-to-Vehicle Communication in Various Road Settings Through Field Trials. In: The 4th IEEE LCN Workshop On User Mobility and Vehicular Networks (ON-MOVE), Denver, CO, USA, pp. 613-620, Oct.. IEEE (2010)
5. Jonsson, M., Kunert, K., Böhm, A.: Increasing the Propability of Timely and Correct Message Delivery in Road Side Unit Based Vehicular Communication. In: Intelligent Transportation Systems Conference (ITSC 2012), Anchorage, AK, USA, Sept.. IEEE (2012)
6. Fernandes, P., Nunes, U.: Platooning with DSRC-Based IVC-Enabled Autonomous Vehicles: Adding Infrared Communication for IVC Reliability Improvement. In: Intelligent Vehicle Symposium, Alcalá de Henares, Spain, June. IEEE (2012)
7. Mak, T.K., Laberteaux, K.P., Sengupta, R.: A Multi-Channel VANET Providing Concurrent Safety and Commercial Services. In: Proc. 2nd ACM International Workshop on Vehicular Ad Hoc Networks (VANET '05), New York, NY, USA, pp. 1-9, Sept.. ACM (2005)
8. Barradi, M., Hafid, A.S., Gallardo, J.R.: Establishing Strict Priorities in IEEE 802.11p WAVE vehicular networks. In: GLOBECOM'10, Miami, FL, USA, pp.1-6, Dec.. IEEE (2010)
9. Bilstrup, K., Uhlemann, E., Ström, E.G., Bilstrup, U.: On the Ability of the 802.11p MAC Method and STDMA to Support Real-Time Vehicle-to-Vehicle Communications. EURASIP J. on Wireless Comm. and Networking, Article number 902414, pp. 1-14 (2009)
10. Butt, M.M.: Provision of Guaranteed QoS with Hybrid Automatic Repeat Request in Interleave Division Multiple Access Systems. In: IEEE Singapore Int. Conference on Communication Systems (ICCS '06), Singapore, pp. 1-5, Oct.. IEEE (2006)
11. Giancola, G., Falco, S., Di Benedetto, M.G.: A Novel Approach to Error Protection in Medium Access Control Design. In: 4th Int. IEEE Workshop on Mobile and Wireless Communication Networks (MWCN '02), Stockholm, Sweden, pp. 337-341, Sept.. IEEE (2002)
12. Fan, X., Jonsson, M., Jonsson, J.: Guaranteed Real-Time Communication in Packet-Switched Networks with FCFS Queuing. Comp. Netw. 53, no. 3, Feb., pp. 400-417 (2009)
13. Kosch, T., Kulp, I., Bechler, M., Strassberger, M., Weyl, B., Lasowski, R.: Communication Architecture for Cooperative Systems in Europe. IEEE Communications Magazine, vol. 47, no. 5, pp. 116-125, May (2009)
14. Pereira, P. P., Casaca, A., Rodrigues, J. J. P. C., Soares, V. N. G. J., Triay, J., Cervelló-Pastor, C.: From Delay-Tolerant Networks to Vehicular Delay-Tolerant Networks. IEEE Communications Surveys & Tutorials, vol. 14, no. 4, pp. 1166-1182 (2012)
15. Jarupan, B., Ekici, E.: A survey of cross-layer design for VANETs. Ad Hoc Networks, vol. 9, pp. 966-983 (2011)

16. Böhm, A., Jonsson, M.: Real-Time Communication Support for Cooperative, Infrastructure-Based Traffic Safety Applications. *Int. J. of Vehicular Technology*, vol. 2011, Article ID 541903, 17 pp (2011)
17. Böhm, A., Jonsson, M.: Supporting Real-Time Data Traffic In Safety-Critical Vehicle-To-Infrastructure Communication. In: *The 2nd IEEE LCN Workshop On User MObility and VEhicular Networks (ON-MOVE)*, Montreal, Canada, Oct. 14, pp. 614-621. IEEE (2008)
18. Jonsson, M., Bergenheim, C., Olsson, J.: Fiber-Ribbon Ring Network With Services For Parallel Processing And Distributed Real-Time Systems. In: *Proc. ISCA 12th International Conference on Parallel and Distributed Computing Systems (PDCS-99)*, Fort Lauderdale, FL, USA, Aug. 18-20, pp. 94-101. ISCA (1999)
19. Liu, C.L., Layland, J.W.: Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment", *J. of the ACM*, vol. 20, no. 1, Jan., pp. 46-61 (1973)
20. Kunert, K., Uhlemann, E., Jonsson, M.: Predictable Real-Time Communications with Improved Reliability for IEEE 802.15.4 based Industrial Networks. In: *Proc. 8th IEEE International Workshop on Factory Communication Systems (WFCS 2010)*, Nancy, France, May 18-21, pp. 13-22. IEEE (2010)
21. Kunert, K., Jonsson, M., Uhlemann, E.: Exploiting Time and Frequency Diversity in IEEE 802.15.4 Industrial Networks for Enhanced Reliability and Throughput. In: *Proc. 15th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2010)*, Bilbao, Spain. 9 pages. Sept. 13-16. IEEE (2010)
22. Jonsson, M., Kunert, K.: Towards Reliable Wireless Industrial Communication with Real-Time Guarantees. *IEEE Tr. on Ind. Informatics*, vol. 5, no. 4, Nov., pp. 429-442 (2009)
23. Jonsson, M., Kunert, K.: Reliable Hard Real-Time Communication In Industrial And Embedded Systems. In: *Proc. third IEEE International Symposium on Industrial Embedded Systems (SIES 2008)*, Montpellier, France, pp. 184-191, June 11-13. IEEE (2008)
24. Hoang, H., Jonsson, M.: Switched Real-Time Ethernet in Industrial Applications - Deadline Partitioning. In: *9th Asia-Pacific Conference on Communications (APCC '03)*, Penang, Malaysia, vol.1, pp. 76-81, Sept.. IEEE (2003)
25. Spuri, M.: Analysis of Deadline Scheduled Real-Time Systems. Technical Report No. 2772, INRIA, France (1996)
26. Baruah, S.K., Mok, A.K., Rosier, L.E.: Preemptively Scheduling Hard-Real-Time Sporadic Tasks on One Processor. In: *IEEE Real-Time Systems Symposium (RTSS '90)*, Lake Buena Vista, FL, USA, pp. 182-190, Dec.. IEEE (1990)
27. Baruah, S.K., Rosier, L.E., Howell, R.R.: Algorithms and Complexity Concerning the Preemptive Scheduling of Periodic, Real-Time Tasks on One Processor. *Real-Time Systems*, vol. 2, no. 4, pp. 301-324 (1990)
28. Stankovic, J.A., Spuri, M., Ramamritham, K., Buttazzo, G.C.: *Deadline Scheduling for Real-Time Systems – EDF and Related Algorithms*. Kluwer Academic Publishers, Boston, MA, USA (1998)