This document is a preprint of: S. Kuhlmorgen, A. Gonzalez, A. Festag et al., “Improving Communication-based Intersection Safety by Cooperative Relaying with Joint Decoding,” in Proceedings of IEEE Intelligent Vehicles Symposium (IV2015 2017), Los Angeles, Jun 2017. DOI:10.1109/IVS.2017.7995796

Improving Communication-based Intersection Safety by Cooperative Relaying with Joint Decoding

Sebastian Kühlmorgen*, Arturo González*, Andreas Festag†, Gerhard Fettweis*
*Vodafone Chair Mobile Communications Systems, Dresden University of Technology, Germany
{sebastian.kuehlmorgen,arturo.gonzalez,gerhard.fettweis}@tu-dresden.de
†Fraunhofer Institute for Transportation and Infrastructure Systems IVI, Dresden, Germany
andreas.festag@ivi.fraunhofer.de

***Abstract*— Vehicular communications have a great potential to improve intersection safety and traffic efficiency. Achieving a high application performance is challenging due to the specific propagation conditions caused by buildings and obstacles found at urban intersections. Relying on the state-of-the-art solution for vehicular communication based on IEEE 802.11, we extend contention-based forwarding to distribute data packets via multiple paths and apply joint decoding on erroneous received data packets. We study the gain of cooperative relaying with joint decoding on the performance of collision avoidance applications in an intersection scenario. We could show that with our algorithm the awareness distance and reliability is increased up to 25 m and 55 %, respectively, under poor channel conditions.**

I. Introduction

Intersections account for more than 20 % of road fatalities in Europe [1], although they represent only a small part of the overall road system. Intersections can easily become bottlenecks for vehicular traffic flows caused by other vehicles, pedestrians, cyclists and tramways, arriving from different directions. Vehicular communications systems (VCS) have a great potential to improve intersection safety and traffic efficiency, including also non-signalized intersections.

Use cases for VCS-based intersection safety cover the prevention of turning and crossing-path collisions, rear-end collisions and traffic light violations; applications for intersection efficiency optimize the traffic flows, e.g., by green light optimized speed advisory (GLOSA), traffic signal adaptation for emergency warning and prioritized road users [2]. Also, VCS greatly enhance vehicle automation and integrate automated vehicles into the overall transport system. Finally, VCS enable advanced methods for cooperative intersection management, such as time slots and space reservations, trajectory planning, and virtual traffic lights [3].

The state-of-the-art solution for VCS relies on a variant of IEEE 802.11 customized for vehicular environments and operating in the 5.9 GHz frequency band allocated for safety and traffic applications, whereas dedicated protocols realize networking and transport, messages for application support, security and management [4]. Specifically, the European system, also referred to as Cooperative Intelligent Transport Systems (C-ITS), applies ad hoc routing and messaging protocols for periodic and event-driven safety information, i.e., Cooperative Awareness Message (CAM) and Decentralized Environmental Notification Message (DENM) [4].

Vehicular safety applications rely on the vehicle’s awareness of its surrounding. A high *awareness probability* [5] implies the reliable and timely exchange of information that meets the application constraints. In the context of intersection safety, collision avoidance applications require a large *awareness distance* in which vehicles receive information from other road users that approach or cross the intersection. Achieving a high awareness distance at intersections is challenging because of the fading effects caused by multi-path propagation and shadowing effects, which impair the communication quality.

In order to extend the awareness range in VCS-based intersection safety applications we exploit that data packets are forwarded over multiple paths and jointly decoded at the receiver. The approach of cooperative relaying with joint decoding (*CR+JD*) [6], also known as cooperative relaying with distributed turbo coding, is applied to C-ITS. While in conventional VCS, received packets with bit errors are discarded, in the considered approach receivers can correctly decode the information from multiple erroneous copies of the same packet received via different paths. Specifically, we extend contention-based forwarding (*CBF*) for multi-path distribution and study the performance gain in an intersection safety scenario.

VCS for intersection safety have been addressed in previous studies, e.g., in [7]–[11]. Authors in [7] presented results from radio channel measurements at different types of urban intersections. Authors in [8] studied the impact of non-line-of-sight (NLOS) propagation on the VCS performance. Authors in [9] analyzed timings in VCS-based collision avoidance systems, [10] optimized VCS messaging protocols to reliably predict potential intersection collisions and [11] investigated platooning at intersections. These studies relied on the state-of-the-art VCS and tuned the communication parameters, respectively. To the best of the authors’ knowledge, the present paper is the first that considered VCS enhanced by *CR+JD*. Additionally, previous publications about *CR+JD*, e.g., [6], analyzed performance gains theoretically using simplified communication models without considering realistic protocols and use cases.

The remainder of this paper is structured as follows: Sec. II gives an overview about cooperative relaying with joint decoding in VCS and describes the packet forwarding algorithms with their enhancements. Sec. III introduces the evaluation scenario and key performance indicators. Sec. IV presents the results of the simulation-based performance evaluation. Finally, Sec. V draws conclusions.

II. Concepts and Algorithms

A. Cooperative Relaying with Joint Decoding in VCS

Cooperative relaying (CR) is a well-known technique that improves reliability and robustness in wireless communication networks.[1] In principle, with cooperative relaying a message is transmitted from a source node via multiple network paths using relays to the destination. The destination processes copies of the same packet, received via different paths, in order to decode the original packet error free. The different paths ensure that fading effects and resulting bit errors are uncorrelated.

Joint decoding (JD) can be regarded as a specific realization of cooperative relaying. It enables the parallel processing of multiple copies of the same packet which was issued by a common source but its copies are affected by independent propagation conditions on different network paths. Joint decoding increases the probability to obtain an error free packet at the destination.

The performance gain of joint decoding can be attributed to the information-theoretic concept of distributed source coding (DSC): the Slepian-Wolf theorem [12] has shown that separate coding is as efficient as joint coding.[2] In the last years, DSC has been applied in more practical settings. For example, [13] proposes an accumulator-assisted distributed turbo code that can be used for a joint decoder. In their scenario, the source broadcasts a packet to a relay and the destination. The relay forwards the packet to the destination, even though the packet contains errors after decoding. Eventually, joint decoding can utilize the corrupted packet from the relay node to recover the original packet at the destination.

In our previous work, we have investigated cooperative relaying with joint decoding in a wireless ad hoc network [14]. This study investigated a WLAN-like system in a simple static scenario with one source, three relays and one destination, where unicast packets were sent from the source to the destination. Also, we considered the effect of forwarding corrupted packets at relay nodes ('lossy forwarding'). With our proposed algorithm CBGF we could improve the packet success ratio by $42\,\mathrm{p.p.}$ and the end-to-end delay by $44\,\mathrm{ms}$, which corresponds to the reliability and latency, respectively.

In this paper we extend our work in [14] by considering broadcast communications, specifically in vehicular communication scenarios. Compared to [14], we exclude the relaying of erroneous packets because in [15] we could show that forwarding of error-free packets yields better performance results than lossy forwarding. This effect comes from the fact that in VCS relays are located on roads and therefore relatively close. If one of the relays receives an error-free packet, it distributes this packet to the other relays in its vicinity. Consequently, lossy forwarding results in inappropriate re-transmissions.

[1] Other benefits are data rate and energy efficiency, which are not considered in this paper.

[2] Details of this fundamental concept is beyond the scope of this paper.

TABLE I
Considered Forwarding Algorithms

Algorithm	Description
CBF	Standardized contention-based forwarding (CBF) algorithm [16]
CBF+CR	CBF using re-transmissions for multi-path distribution of packets (cooperative relaying)
CBF+CR+JD	CBF using cooperative relaying with joint decoding

In contrast to unicast communication, with broadcast every node is relay and destination at the same time. Hence, all nodes apply joint decoding and collect packet copies as long as they cannot recover the original packet error-free. Once the packet is successfully decoded, it is further processed by the forwarding algorithm.

B. Forwarding Algorithms

This section describes contention-based forwarding (*CBF*) and its extensions for multi-path routing with joint decoding. *CBF* is one of the forwarding algorithms standardized in C-ITS [16]. It relies on two key mechanisms, i.e., timer-based re-transmissions and overhearing. With *CBF*, when a sender broadcasts a packet to its neighbors, the receivers buffer the packet and start a timer that depends on the distance to the sender – the longer the distance, the shorter the timer duration – to ensure a large forwarding progress. When the first timer expires (at the farthest neighbor), the packet is re-broadcasted. This packet is overheard by surrounding vehicles, which stop their timer and discard their packet. Thus, overhearing is relevant for the efficiency of the forwarding algorithm since only one packet is re-transmitted, resulting in a single path.

In order to apply joint decoding at the destination multiple copies of the original packet (identified by common source ID and packet sequence number) are needed. In plain *CBF* just a single path is available and JD would not be feasible. For this reason we extend *CBF* by multi-path distribution, which can be achieved by the introduction of multiple re-transmissions. To avoid an uncontrolled flooding of the network an overhearing retransmission counter (RC) and threshold (RT) is used to limit parallel paths. Every time when a copy of the packet (a duplicate) is received RC is incremented, which is proceeded up to RT is reached. Then all already received and new arriving packet copies will be dropped. All nodes only forward one packet copy (same source ID and sequence number). We denote this extension by *CBF+CR* without joint decoding and *CBF+CR+JD* with joint decoding.

In ***CBF+CR+JD*** whenever a node receives a packet copy, and the previous received copies were corrupted, it executes joint decoding including all collected packet copies which are stored in a PHY buffer (see Fig. 1, access layer). A successful decoded packet is passed to the networking layer, which includes the duplicate detection list, the overhearing mechanism

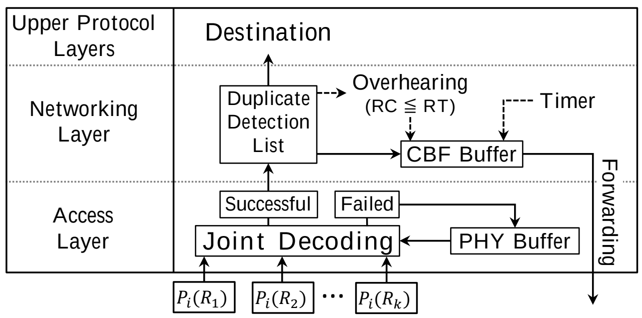

Fig. 1. Shown is the *CBF+CR+JD* algorithm. $P_i(R_k)$ denotes packet i, received from relay k, i.e., we have k packet copies. Solid lines depict packet flow, and dashed lines depict control information.

with the extended RC and RT, the timer computation and the CBF buffer. The first successful received packet is logged in the duplicate detection list and is passed directly to upper layers. Further steps are: the packet is buffered, the timer is started, and the overhearing mechanism increments RC. If the timer expires and RC $\leq$ RT, the packet is forwarded (see TABLE II). Otherwise, it is dropped.

For our investigations in this paper we compare three algorithms shown in TABLE I. The first two, *CBF* and *CBF+CR*, are considered as baselines and we compare their performance with our enhanced algorithm *CBF+CR+JD*, the variant extended by joint decoding.

III. Evaluation Scenario and Key Performance Indicators

A. Evaluation Scenario

We consider a VCS in a non-signalized, cross-shaped road intersection scenario with vehicles that approach the intersection and use a collision avoidance application. All vehicles are assumed to be equipped with communication capabilities and to execute the C-ITS protocol stack [4]. Every vehicle generates DENMs of 1,000 bytes with a message generation time interval that is varied between 0.1 and 1 s for our simulations. Among other information, these DENMs contain driving direction, speed, generation time of the message, position of the vehicle, relevance area and DENM expiry time. The expiry time limits the duration in which a message is retransmitted.

The considered collision avoidance application uses the DENM and CBF protocols to distribute collision warnings within the relevance area. Compared to the single-hop CAM,[3] a DENM can be transmitted over multiple wireless hops, which is beneficial to enlarge the relevance area beyond a node's communication range, specifically in intersection scenarios with NLOS communication. DENMs also facilitate re-transmissions and multi-path distribution of packets. Both features, re-transmissions and multi-path distribution, enable the cooperative relaying with joint decoding. A summary of simulation parameters can be found in TABLE II.

In the receiving vehicles, the successfully decoded DENMs are passed to the safety applications, i.e., to the collision

[3] CAM and DENM are standardized in ETSI EN 302 637-2/ -3.

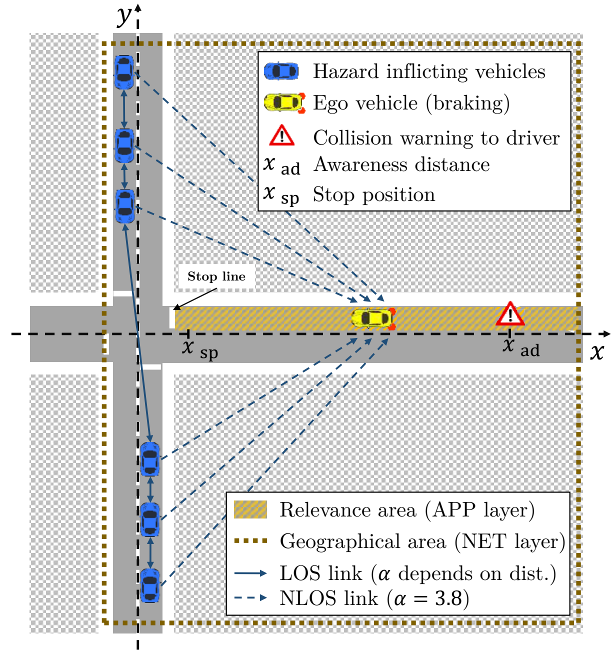

Fig. 2. Intersection scenario with vehicles on collision course. The ego vehicle receives a collision warning and executes a full breaking.

avoidance application in our scenario. The message is only passed to the application if the vehicle is located inside a relevance area, which is determined by the source node's application and included in the DENM. When the collision avoidance application detects a potential collision of the ego-vehicle with another vehicle, it warns the driver, who may then execute appropriate driving maneuvers, such as full braking. The relevance area restricts the geographical area for the application, and messages outside will be discarded.

The scenario is illustrated in Fig. 2 from the perspective of the vehicle that approaches the intersection on the horizontal road and is denoted as ego-vehicle: vehicles drive on the vertical road in both directions[4] and have priority over the ego-vehicle. The vertically driving vehicles are not on collision course with each other and the warnings are not shown to their drivers. Still, these vehicles are important for relaying of DENMs because re-transmissions and multi-path communication is required for cooperative relaying with joint decoding of packets at the ego-vehicle.

Specifically, in our scenario the relevance area is defined by the width of the road and has a length of 400 m originating from 5 m (stop line) before the intersection. When the ego vehicle approaches the intersection, it enters the relevance area. We assume an ego-vehicle speed of 80 km/h, which represents an upper speed limit. The channel model differentiates between LOS for the vertically driving vehicles and NLOS for the links between the ego vehicle and the other (vertically driving) vehicles. The latter is indicated by the grid areas (representing forest or signal blocking obstacles, such as buildings) in Fig. 2. For LOS and NLOS, we apply different

[4] For simplicity, we assume that the vehicles on the vertical road do no turn.

TABLE II
SIMULATION PARAMETERS

Parameter	Values
Number of vehicles	7
ITS-G5 PHY	PHY link level abstraction [18]
ITS-G5 MAC	IEEE 802.11 DCF with CSMA/CA
Forwarding algorithm	CBF with enhancements (see Sec. II-B)
CBF parameter	RT = 5, Max timeout = 100 ms
Data rate	6 Mbit/s
Transmit power	23 dBm
Channel bandwidth	10 MHz at 5.9 GHz
DENM time interval	$T_{\text{int}} = 0.1 - 1.0\,\text{s}$
DENM size	1,000 bytes
Speed limit	22.2 m/s (80 km/h or 50 mph)
Driver reaction time t_r	1.3 s [19]
Deceleration vehicle	$5.886\,\text{m/s}^2$ [19]
Simulation time	32 s
Simulation runs (seeds)	10,000

TABLE III
CHANNEL MODEL PARAMETERS

Path loss model	
LOS	NLOS
Two-Log-Distance (d) $\alpha_1 = 2.1$ ($d \leq 100\,\text{m}$) $\alpha_2 = 3.8$ ($d > 100\,\text{m}$)	Log-Distance (d) $\alpha = 3.8$
Fading model	
LOS	NLOS
Nakagami-m fading $m_1 = 1.52$ ($d \leq 90.5\,\text{m}$) $m_2 = 0.74$ ($d \leq 230.7\,\text{m}$) $m_3 = 0.84$ ($d > 230.7\,\text{m}$)	Nakagami-m fading $m = 0.84$

path loss models with Nakagami-m fading (see TABLE III for an overview of the model parameters). All parameters are taken from [17], data set 1, and validated by measurements in real-world suburban driving conditions. The reason to use a two-log distance pathloss model and three distance Nakagami-m fading for the LOS is that with greater distance on a highway vehicles can block the link between two vehicles.

The simulations were performed with the network simulator *ns-3* for data traffic and *SUMO* the road traffic. In *ns-3*, we have used the *WAVE* module for vehicular communications and extended it by a PHY link level abstraction for joint decoding [18], *CBF*, *CBF+CR* for cooperative relaying, and DENMs for collision warnings. The PHY link level abstraction is restricted to three packet copies. To achieve realistic results, *ns-3* was coupled bidirectionally, based on the *TraCI* C++ API, with the road traffic simulator *SUMO*. The coupling realizes a mapping between *ns-3* nodes and *SUMO* vehicles. For every vehicle entering the *SUMO* simulation a new *ns-3* node is created and initialized. The positions of *ns-3* nodes are updated every period with the new information retrieved from *SUMO*. In our scenario when the ego vehicle received the first collision warning, and the driver applies a full braking the new speed is sent back into *SUMO*, which adapts the simulation parameters.

B. Key Performance Indicators

For comparison of the three algorithms we define four key performance indicators (KPIs) to evaluate the reliability of

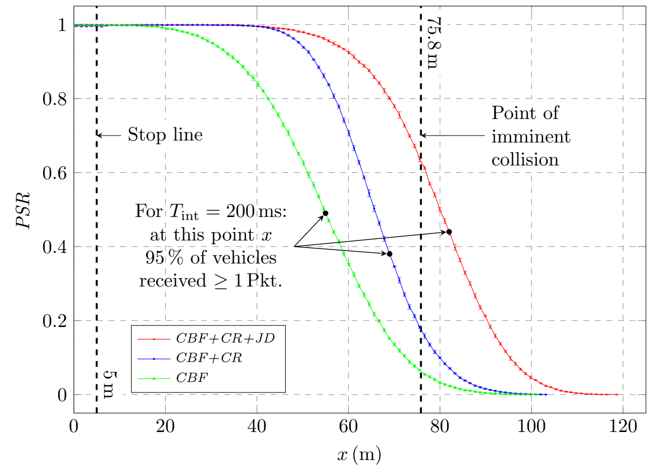

Fig. 3. Packet success ratio of the ego vehicle.

packet reception, the position of the first warning, the stop position, and the number of successfully received packets during braking, split in different decoding attempts.

Packet Success Ratio (PSR) describes the probability of packet reception dependent on the distance. For instance, if 100 packets are sent by all vertically driving vehicles and 87 of them will be received successfully at the ego vehicle, we have a PSR of 0.87. In our case, the PSR refers only to the ego vehicle because we are interested in the performance of the collision avoidance application. It is worth recalling that the communication relies on broadcast even if we focus on the ego vehicle. With the PSR and the DENM time interval (T_{int}), we are able to derive the *awareness probability* as defined in [5], i.e., the probability to receive at least n packets error-free in a given time window. For the collision avoidance application we need at least one packet ($n \geq 1$).

Awareness Distance (AD) is the distance from the position x_{ad} (see Fig. 2), when the ego vehicle receives the first collision warning (DENM), to the center of the intersection. When the first message is received, the applications can trigger a warning to the driver, or in automated vehicles take over control.

Stop Position (SP) at position x_{sp} (see Fig. 2) is the position when the ego vehicle stops completely. $x_{\text{sp}} = 0$ is at the center point of the intersection. Stop lines of roads are 5 m away from the center of the intersection, which should be the last position to stop the vehicle safely.

Decoding Attempts (DA) represents the number of packet copies that are needed to decode the original packet error-free. The packet copies are created through relaying and multi-path distribution towards the destination vehicle (in our broadcast scenario every vehicle is destination and relay at the same time). For instance, if the destination is not able to decode the first received packet error-free it needs to wait for at least one more packet copy. In conventional decoding the vehicle would discard the first and try decoding the second (or subsequently) received packet, whereas with joint decoding several packet copies are decoded jointly.

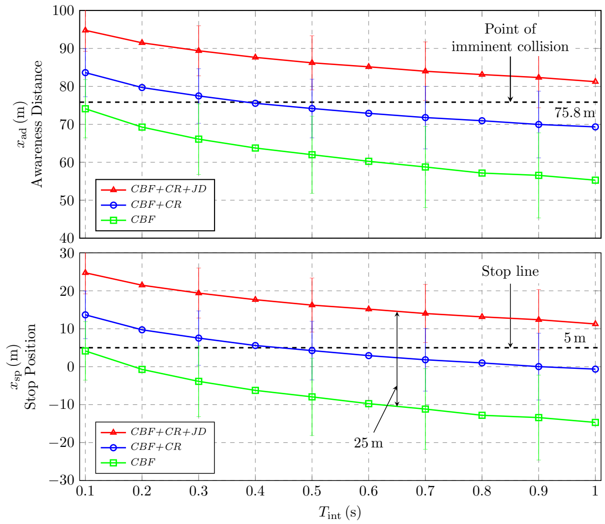

Fig. 4. Shown is the averaged awareness distance, which is the position of the first received error-free packet, (top) and stop position (bottom) of the ego vehicle. The graph lines should not fall below the dashed lines, otherwise we have an imminent collision.

IV. Performance Evaluation

As a first step, we determine the averaged *PSR* for the approaching ego vehicle (Fig. 3). For this, vehicles were stopped at particular points in time in the simulation and then every vertical vehicle sent 1,000 packets. These simulation runs were performed with 10 different seeds for each position and vehicle. In Fig. 3 the dashed lines at 5 m and 75.8 m represent the stop line and point of imminent collision (PIC), respectively (see also Fig. 2 for the overall scenario). If a vehicle would pass the PIC, it crosses the intersection and causes an accident. The algorithm using cooperative relaying with joint decoding exhibits a *PSR* above 60 % at the PIC. As shown in Fig. 4, with this *PSR* no vehicle crosses the stop line on average for all time intervals. At the PIC, *CBF* and *CBF+CR* perform with a *PSR* less than 20 % and 10 %, respectively. Therefore, for longer time intervals the ego vehicle is not able to stop at the right time. The highlighted black dots in Fig. 3 indicate the value of x, where 95 % of vehicles have received at least one packet for a time interval of 200 ms. For instance, for *CBF+CR+JD* 95 % of vehicles have received at least one packet passing a distance of 82 m towards the intersection. Clearly, *CBF+CR+JD* shows a much better *PSR* compared to the other two algorithms.

In the upper part of Fig. 4 the averaged awareness distance x_{ad} (i.e., the distance when the ego vehicle's application receives the first packet and applies full braking), for different values of the message time interval T_{int} is shown. This is complemented by the stop position x_{sp} over T_{int} in the bottom part of Fig. 4. In principle, none of the values of x_{ad} and x_{sp} should be below the dashed lines in the figures, otherwise an accident would occur. We can see that for message time intervals longer than 650 ms, *CBF+CR+JD* has a gain of 25 m in distance compared to the *CBF* and 12 m compared to *CBF+CR*. On average, *CBF+CR+JD* stays always above the dashed lines, whereas *CBF* stays always below the dashed

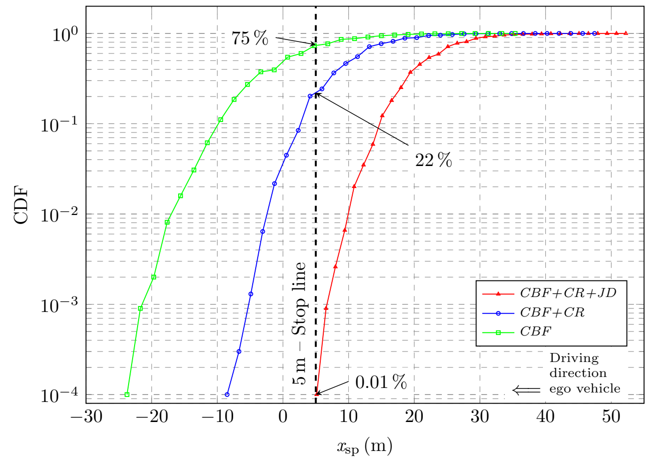

Fig. 5. Cumulative density function of ego vehicles' stop positions for a DENM time interval of 200 ms. The graphs can be interpreted as the portion of vehicles that have not stopped to this x_{sp}. The ego vehicles should completely stop before the stop line (dashed line) to refrain from affecting vehicles on the priority road (vertical road Fig. 2).

line and is therefore unsuitable in this intersection scenario. *CBF+CR* is only useful for low message time intervals; for intervals longer than $T_{int} = 400$ ms, x_{sp} is below the stop line.

In Fig. 4, the graphs of all algorithms decrease with a growing DENM time interval. With the given speed of the ego-vehicle, it needs approximately the same time to cross the intersection. During this time, the shorter the message time interval gets, the more packets the vertical vehicles can send, which leads to an increasing probability of packet reception and therefore a higher awareness probability (see [5]). Another advantage of *CBF+CR+JD* compared to *CBF+CR* is the possibility to decrease the DENM time interval in order to spare channel resources. In future vehicular networks, when the equipment rate of vehicles grows up to 100 %, congestion control will play a major role. Therefore, it is beneficial to reduce the message rate as much as possible without compromising the safety level.

To illustrate the feasibility of ego vehicles for a full stop before the stop line we show the cumulative density function (CDF) of stop positions x_{sp} for the ego vehicle in Fig. 5. The CDF can be interpreted as the portion of vehicles that have not stopped to this x_{sp}. With the given speed of vehicles, the generation rules for periodic messages in C-ITS[5] would determine a message time interval of approximately 200 ms. For the sake of comparability we therefore set $T_{int} = 200$ ms. For other DENM time intervals, we have verified that the results are equivalent, i.e., shifted on the x-axis. Again, the dashed line depicts the stop line, which is 5 m from the center point of the intersection. We can observe that for *CBF+CR+JD* only a single vehicle reaches the stop line, which correspond to 0.01 % (using 10,000 different seeds). The other two algorithms, *CBR* and *CBR+CR*, show a significant worse performance in terms of avoiding the risk of imminent collisions. For *CBF*, 75 % of vehicles exceed the

[5]Originally, these rules were defined for CAM and we apply the same for DENMs in this study.

TABLE IV
PERCENTAGE OF ERROR-FREE RECEIVED PACKETS FOR THE EGO VEHICLE NORMALIZED TO *CBF*, I.E., *CBF* RECEIVES 100 % OF PACKETS.

Algorithms	Total Rx packets (%)	Decoding attempts (%) 1	2	3
CBF	100.0	90.93	8.60	0.47
CBF+CR	128.7	91.71	19.87	17.12
CBF+CR+JD	155.3	89.13	40.92	25.25

stop line. For *CBF+CR*, 22 % cause an accident, which is a better result than *CBF*, but still does not meet the safety requirements.

TABLE IV shows the total ratio of successfully received packets and the ratio of different decoding attempts 1, 2 and 3, normalized to results of *CBF*. The number of received packets are only counted inside the relevance area, and the maximum number of packet copies used for joint decoding is three because of the restricted PHY link level abstraction. We can see that with *CBF+CR+JD* 55 % and with *CBF+CR* 29 % more packets can be decoded error-free than with the conventional *CBF*. The other three columns in TABLE IV split the total ratio of successfully decoded packets into the number of decoding attempts. For example, two decoding attempts means that for conventional decoding (*CBF* and *CBF+CR*) the first packet was dropped because of errors after decoding and the second packet was received error-free. For *CBF+CR+JD* joint decoding is used, which means the first packet is not dropped but buffered. Upon reception of the second packet copy, both packets are used in the decoding process and result in an error-free packet. Overall, with joint decoding many more packets can be decoded error-free, which is not only critical for collision avoidance applications in intersection scenarios. In particular, applications that rely on the transmission of a sequence of packets or a bi-directional exchange of packets, e.g., lane change assistance or platooning applications can benefit.

V. CONCLUSION

In this paper we evaluated the performance of a VCS enhanced by cooperative relaying and joint decoding. We considered an intersection scenario with poor channel conditions and high vehicle speeds and assessed the performance of a collision avoidance application. We have presented extensions of the standardized protocol stack for VCS by cooperative relaying with joint decoding with the objective to extend the communication range and the awareness probability. Comparing the extended with the standardized system, we could show that cooperative relaying with joint decoding increases the awareness distance by 25 m and enables receiving 55 % more packets error-free inside the communication range. In the considered scenario, in all cases the vehicles could always stop before the stop line, whereas only 25 % of the vehicles met this requirement using the conventional system. Though our scenario should be considered as a worst case giving a high performance gain. The results indicate that the advanced communication techniques can help improving VCS-based intersection safety, where it is of high importance to inform the driver or future automated vehicles as early as possible to let them take timely actions.

ACKNOWLEDGMENT

This work has been supported by the German Federal Ministry of Education and Research within the program “Twenty20” under contract 03ZZ0513J - “fast traffic”.

REFERENCES

[1] European Commission, “Traffic Safety Basic Facts 2016: Junctions,” Technical Report, European Commission, Directorate General for Transport, Jun. 2016.
[2] L. Le, A. Festag, R. Baldessari, and W. Zhang, “Vehicular Wireless Short-range Communication for Improving Intersection Safety,” *IEEE Communications Magazine*, vol. 47, no. 11, pp. 104–110, Nov. 2009.
[3] L. Chen and C. Englund, “Cooperative Intersection Management: A Survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 570–586, Feb. 2016.
[4] A. Festag, “Cooperative Intelligent Transport Systems Standards in Europe,” *Communication Magazine*, vol. 12, no. 52, Dec. 2014.
[5] N. An, T. Gaugel, and H. Hartenstein, “VANET: Is 95% Probability of Packet Reception Safe?” in *2011 11th International Conference on ITS Telecommunications (ITST 2011*, Saint-Petersburg, Russia, Aug. 2011, pp. 113–119.
[6] K. Anwar and T. Matsumoto, “Accumulator-Assisted Distributed Turbo Codes for Relay Systems Exploiting Source-Relay Correlation,” *IEEE Communications Letters*, vol. 16, no. 7, pp. 1114–1117, Jul. 2012.
[7] J. Karedal *et al.*, “Radio Channel Measurements at Street Intersections for Vehicle-to-Vehicle Safety Applications,” in *IEEE VTC Spring 2010*, Taipei, Taiwan, May 2010, pp. 1–5.
[8] T. Mangel, F. Schweizer, T. Kosch, and H. Hartenstein, “Vehicular Safety Communication at Intersections: Buildings, Non-Line-Of-Sight and Representative Scenarios,” in *IEEE WONS 2011*, Bardonecchia, Italy, Jan. 2011, pp. 35–41.
[9] A. Tang and A. Yip, “Collision Avoidance Timing Analysis of DSRC-based Vehicles,” *Accident Analysis & Prevention*, vol. 42, no. 1, pp. 182–195, 2010.
[10] S. Joerer *et al.*, “A Vehicular Networking Perspective on Estimating Vehicle Collision Probability at Intersections,” *IEEE Transactions on Vehicular Technology*, vol. 63, no. 4, pp. 1802–1812, May 2014.
[11] H. J. Günther, S. Kleinau, O. Trauer, and L. Wolf, “Platooning at Traffic Lights,” in *IEEE Intelligent Vehicles Symposium (IV)*, Göteborg, Sweden, June 2016, pp. 1047–1053.
[12] D. Slepian and J. Wolf, “Noiseless Coding of Correlated Information Sources,” *Information Theory, IEEE Transactions on*, vol. 19, no. 4, pp. 471–480, Jul. 1973.
[13] K. Anwar and T. Matsumoto, “Accumulator-assisted Distributed Turbo Codes for Relay Systems Exploiting Source-Relay Correlation,” *IEEE Communications Letters*, vol. 16, no. 7, p. 1114–1117, Jul. 2012.
[14] S. Kühlmorgen, A. Festag, and G. Fettweis, “Exploiting Distributed Source Coding for Multi-Hop Routing in Wireless Ad Hoc Networks,” in *IEEE VTC Spring 2016)*, Nanjing, China, May 2016, pp. 1–6.
[15] ICT-619555 Deliverable, “D3.2 – Report on Revised WP3 Architecture Including Simulation Results,” RESCUE, Tech. Rep., Jul 2015.
[16] ETSI, *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 (ETSI EN 302 636-4-1 V1.2.1)*, Jul. 2014.
[17] L. Cheng, B. E. Henty, D. D. Stancil, F. Bai, and P. Mudalige, “Mobile Vehicle-to-Vehicle Narrow-Band Channel Measurement and Characterization of the 5.9 GHz Dedicated Short Range Communication (DSRC) Frequency Band,” *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1501–1516, Oct. 2007.
[18] V. Tervo *et al.*, “An Error Rate Model of Relay Communications with Lossy Forwarding and Joint Decoding,” in *ICC 2016*, Kuala Lumpur, Malaysia, May 2016, pp. 528–533.
[19] S. J. Brunson, E. M. Kyle, N. C. Phamdo, and G. R. Preziotti, “Alert Algorithm Development Program NHTSA Rear-End Collision Alert Algorithm Final Report,” National Highway Traffic Safety Administration U.S. Department of Transportation, Tech. Rep. HS 809 526, Sep. 2002.