Mitigating Dead Spots in Cellular Networks with the Hybrid Communication Protocol CAMFLOOP

Ergys Puka and Peter Herrmann[1]

Abstract— **The interaction between vehicles and the external infrastructure usually relies on the availability of cellular networks. Large sparsely populated regions, however, contain often extended areas with missing cellular network connectivity, so-called *dead spots*. To accelerate the delivery of messages from a vehicle to the infrastructure in such zones, we developed a *Context-Aware Message Flooding Protocol* that utilizes ephemeral ad-hoc networks between nearby vehicles in a dead spot. This allows us to let a message be delivered by an ad-hoc network peer instead of its creator if the peer leaves the dead spot earlier. In previous work, we introduced an initial version of this protocol and proved formally that it guarantees the fastest possible delivery of messages via the cellular network to the infrastructure, and, at the same time, keeps the number of produced duplicates to a minimum. This version was, however, based upon idealized drivers' behavior since we assumed that a vehicle leaves a dead spot exactly at the point of time previously predicted. To rectify this strict assumption, in this paper we present an updated protocol version named CAMFLOOP. Unlike its predecessor, it considers deviations like speed changes or aberrations from the planned route. Moreover, we report on our tests of CAMFLOOP using the traffic simulator SUMO with scenarios inspired by the Australian Outback, a region with dead spots that may span hundreds of kilometers. The simulation results show that deviations from the planned speeds and routes lead to communication errors only rarely. Further, the protocol provides significant reductions of the delivery time by more than 40% on average in larger dead spots, while usually less than two copies of each message created in a dead spot are delivered to the infrastructure.**

I. INTRODUCTION

In Intelligent Transport Systems (ITS), Vehicle-to-Infrastructure (V2I) communication connecting mobile road users with their fixed external infrastructure, usually relies on cellular networks [1]. This technology, however, has the practical issue that it is not available everywhere. In particular, in large and very sparsely populated regions, *dead spots*, i.e., areas without cellular network coverage, can extend several hundred kilometers (see, e.g., the coverage maps provided in [2]). The main reason is that the expensive earthbound cell tower infrastructure is only provided in populated areas and on main roads [3]. Another factor can be mountainous terrain, where hillsides lead to echoes aggravating the reception of radio signals [4].

To reduce the impact of dead spots by accelerating the transmission of messages generated while not having cellular network connectivity, we developed a series of data dissemination protocols [5]–[7]. They combine the cellular network communication with ephemeral Vehicle-to-Vehicle (V2V) networks. Utilizing spatiotemporal properties of the connected vehicles, e.g., their positions, speeds, and planned routes, messages can then be forwarded to the peer of an ad-hoc network, that is predicted to leave the dead spot first. An illustrative scenario, in which our protocols lead to a significant reduction of the delivery time, is when a vehicle, that creates a message shortly after entering a dead spot, hands it off to opposing traffic. Due to its antithetical direction, the recipient is already close to the end of the dead spot and can therefore deliver the message faster via the cellular network than the originator of the message.

Initially, we developed two data dissemination protocols that assign the delivery of a message via the cellular network to exactly one vehicle that, however, can be different from the message creator. The version presented in [5] uses the previous travelling time of the vehicles until meeting each other to determine which one will leave the dead spot first. In contrast, in [6], we introduced a version that utilizes connectivity maps [8] for this purpose. In contrast to its predecessors, the so-called *Context-Aware Message Flooding Protocol* [7] considers that, particularly in large dead spots, vehicles may take part in several ad-hoc networks such that various opportunities to speed-up the forwarding of a message out of the dead spot arise. We proved that this protocol guarantees the quickest possible delivery of messages via the cellular network. This achievement is at the cost of sending duplicates of messages but we could also verify that the number of duplicates is kept to a minimum.

Yet, our proof was based upon the idealized assumption that the vehicles always reach the edges of a dead spot at the previously estimated times. In reality, however, drivers make unscheduled stops, deviate from the initial itinerary specified in their route guidance systems, or use different speeds than expected. Therefore, the idealized assumption of the original protocol [7] is overly restrained. To alleviate this weakness, we created a new version of the Context-Aware Message Flooding Protocol that we call CAMFLOOP. In contrast to the original protocol, each vehicle periodically estimates its dead spot leaving time and adapts the forwarding strategy for messages accordingly, when possible.

CAMFLOOP is introduced in this paper. Moreover, in order to get quantitative data about how well it performs, we simulated its use in various scenarios. For that, we implemented CAMFLOOP at the top of the highly versatile simulator *Simulation of Urban MObility (SUMO)* [9].

While CAMFLOOP can use various Vehicular Ad-Hoc Networks (VANET) and other peer-to-peer networks to

[1]The authors are with the Department of Information Security and Communication Technology, Norwegian University of Science and Technology (NTNU), O.S. Bragstads plass 2, 7491 Trondheim, Norway {puka|herrmann}@ntnu.no

build the ad-hoc connections between passing vehicles, we made good experience with the widespread Device-to-Device (D2D) protocol WiFi Direct [10] in our previous works. Therefore, we implemented WiFi Direct also at the top of SUMO. Further, we created several scenarios based upon a route covering two towns in the Australian Outback. Our extended SUMO variant carried out a vast set of different simulations which allowed us to learn about the strengths and weaknesses of CAMFLOOP. Presenting CAMFLOOP and discussing relevant simulation results are the main contributions of this paper.

The article is structured as followed: After looking at related work in Sect. II, we introduce CAMFLOOP in Sect. III. Thereafter, we present the SUMO extension and the simulated scenarios in Sect. IV. Results obtained from the simulations are then discussed in Sect. V followed by some concluding remarks in Sect. VI.

II. RELATED WORK

Combining cellular and ad-hoc networks to mitigate dead-spots is already used in form of so-called Multi-hop Cellular Networks (MCN) [11]. The authors of [12] provide a good survey of several MCN-based approaches. The main difference to our approach is that MCNs use intermediate peers solely to relay messages but do not store them in phases of being off-line. Thus, they work nicely in small dead spots with a lot of traffic since, in this case, the peers are close enough to find constantly multi-hop connections to the next cellular tower. On country roads, that pass large dead spots and are only used by a few hundred vehicles a day, however, the distance of the peers is usually too large to provide this form of connectivity.

Opportunistic Networks (OppNet) [13] are likely the technology closest to our approach. Unlike our data dissemination protocols, which apply intermediaries just to speed up message delivery, however, OppNets are used when no stable end-to-end connection between the source and a destination of a packet can be established otherwise. The peers of an opportunistic network have also the ability to store packets temporarily and exchange them with other devices via ephemeral ad-hoc connections. In contrast to the well-known Delay-Tolerant Networks (DTN) [14], OppNets do not need a general Internet-like topology or other a-priori knowledge of the network structure [15]. Instead, various routing strategies can be used to decide, in which way packets shall be forwarded in order to make their speedy delivery likely. A survey of routing protocols for OppNets is provided in [16].

Since it is very expensive to test vehicular ad-hoc networks with real vehicles, simulation is a popular means to validate such protocols. In their overview about simulators [17], the authors distinguish between special VANET simulators, network simulators, and mobility generators. While most approaches use network simulators, mobility generators like SUMO are applied less. In [18], SUMO is combined with the network simulator OMNeT++ to simulate a propagation scheme for a cloud-based dissemination scheme for emergency messages between vehicles. The authors of [19] use SUMO in combination with the game engine Unity 3D to simulate and visualize a V2I-based traffic light assistance system. Finally, the approach presented in [20] is based on the self-made traffic simulator ISR-TFS that is applied to simulate a traffic management system. Like us, in these works scenarios of typical traffic flows are used to model several aspects of the used systems.

III. CAMFLOOP

As mentioned above, our protocol CAMFLOOP utilizes the short-range communication capabilities of nearby vehicles in *dead spots* to minimize the transport of messages to areas with sufficient connectivity. When several vehicles $veh_1, \ldots, veh_n$ are connected to an ad-hoc network, their spatiotemporal properties are compared to find out the vehicle that most likely will regain connectivity earlier than all other known peers. Let us assume that veh_1 is the peer in the ad-hoc network, that will leave the dead spot first according to the currently available information. Then the peers check for each message m carried by at least one of them, if any peer is aware of a vehicle veh_o out of our ad-hoc network that carries also a copy of m and will be able to deliver m earlier than veh_1. If that is the case, no vehicle of the ad-hoc network, including veh_1, will be responsible to transmit the message via the cellular network after regaining connectivity since that would be later than the delivery by veh_o. If, however, no peer of the ad-hoc network is aware of such an external vehicle veh_o, then veh_1 is the fastest known dead spot leaver. Therefore, veh_1 commits itself to deliver m immediately after regaining cellular network coverage.

Independently of the fact, whether veh_1 will send m or not, all ad-hoc network peers store a copy of m, if they have not done that already before. If any ad-hoc network member veh_i meets another vehicle veh_t in a later ad-hoc network, that is supposed to leave the dead spot earlier than veh_1 or veh_o respectively, the delivery of message m will be further sped up. In this case, veh_i transmits m to veh_t that delivers the message via the cellular network. Of course, that leads to the delivery of a duplicate since veh_1 or veh_o can normally not be informed about the faster delivery by veh_t anymore. For this reason, the fastest possible delivery of messages can be only guaranteed when such duplicates are tolerated.

The CAMFLOOP protocol stack in a vehicle uses a *transmission buffer* that stores all messages to be immediately sent after regaining cellular network access, while messages, that are just kept to utilize opportunities that might arise in later ad-hoc networks, are kept in the *opportunity buffer*. A vehicle might, however, send a message m also from the opportunity buffer via the cellular network if it, thanks to a higher speed than previously expected, leaves the dead spot earlier than the supposed leaving time of the vehicle previously selected to deliver m. To avoid overflows of the opportunity buffers, each newly created message m is assigned an expiry date by its originator. That shall be a time, at which the message should be already delivered except for the most extreme circumstances. A vehicle may delete m from the opportunity buffer when its expiry date has passed.

To manage the message handling by CAMFLOOP, we provide each message m with additional information:

- *id*: A unique identifier for each created message,
- *tag_time*: The earliest proposed dead spot leaving time of all vehicles committed to deliver m, according to the knowledge of the bearer of m,
- *tag_owner*: The identifier of the vehicle, that is supposed to leave the dead spot at the time stored in tag_time,
- *expiry_date*: The expiry date of m.

In the following, we sketch the functionality of CAMFLOOP by presenting three algorithms in pseudo-code. Algorithm 1 shows the core functions run in each vehicle in relatively short periods. In Algorithm 2, we describe the communication between the vehicles in an ad-hoc network to synchronize each other. Finally, we exemplify the updating of the transmission and opportunity buffers after receiving a synchronization message in Algorithm 3.

Algorithm 1: CAMFLOOP core functionality.

```
1: update_adhoc_members ← false
2: if veh ∉ dead spot then
3:    for all m_ob in opportunity buffer do
4:       if m_ob[tag_time] > current_time then
5:          Store m_ob in transmission buffer.
6:          Remove m_ob from opportunity buffer.
7:       end if
8:    end for
9:    for all m_tb in transmission buffer do
10:      Deliver m_tb to the infrastructure.
11:      m_tb[tag_time] ← current_time.
12:      m_tb[tag_owner] ← veh.
13:      Store m_tb in opportunity buffer.
14:      Remove m_tb from transmission buffer.
15:   end for
16: else
17:   for all m_tb in transmission buffer do
18:      if |m_tb[tag_time] − leaving_time| > δ then
19:         m_tb[tag_time] ← leaving_time.
20:         m_tb[tag_owner] ← veh.
21:         update_adhoc_members ← true.
22:      end if
23:   end for
24:   for all m_ob in opportunity buffer do
25:      if m_ob[tag_time] > leaving_time then
26:         m_ob[tag_time] ← leaving_time.
27:         m_ob[tag_owner] ← veh.
28:         Store m_ob in transmission buffer.
29:         Remove m_ob from opportunity buffer.
30:         update_adhoc_members ← true.
31:      end if
32:   end for
33: end if
34: if veh ∈ ad-hoc network and
    (update_adhoc_members = true or
    network_composition_changed = true) then
35:   synchronize(veh).
36: end if
```

Algorithm 1 coordinates the transmission of messages if the vehicle veh running it, regains cellular network connectivity. Further, when veh is in a dead spot, the algorithm manages the updating of the messages if its supposed leaving time changes significantly. To achieve that, the algorithm is executed in certain periods of, e.g., 30 seconds. In addition, it is carried out when the vehicle leaves a dead spot, a new ad-hoc network has been built up, or the composition of the ad-hoc network has changed due to a vehicle joining or leaving it.

The message handling in regions with cellular network connectivity is described by the pseudo-code between lines 2 and 15. As mentioned above, the vehicle might have left the dead spot earlier than planned. Therefore, it could have been faster than the vehicle assigned to deliver a certain message. To handle this case, each message m_{ob} in the opportunity buffer is checked if its tag time is still in the future. If that is the case, m_{ob} is moved to the transmission buffer to guarantee its fastest delivery (see lines 3 to 8). Thereafter, all messages m_{tb} in the transmission buffer are sent to their recipients via the cellular network as described by the loop between lines 9 and 15. To recall in possible later ad-hoc networks that m_{tb} was indeed sent, a copy of it is stored in the opportunity buffer tagged with the time of delivering m_{tb} and veh as the tag owner.

The lines 17 to 32 handle the update of the proposed leaving times due to unexpected speed changes if veh is in a dead spot. If the leaving time alters by more than a certain threshold δ (usually 30 seconds), CAMFLOOP amends the tag times for the messages in the transmission buffer (see lines 17 to 23). Since the other ad-hoc network members are informed about these changes of the tag time, we use the threshold to avoid unnecessary synchronization runs after only negligible leaving time changes. Thereafter, we check for all messages m_{ob} in the opportunity buffer if their tag times are later than our newly calculated leaving time. In that case, vehicle veh will improve the delivery, and m_{ob} is moved into the transmission buffer (see lines 24 to 32).

When either the tag time of a message in the transmission buffer is amended, a message is copied from the opportunity buffer to the transmission buffer, or the status of the ad-hoc network changes, vehicle veh synchronizes its status with the other members of the ad-hoc network (see lines 34 to 36 of Algorithm 1). The synchronization itself is depicted in Algorithm 2. It assumes that one peer[1] of the ad-hoc network carries out certain managing functionality. Since this *manager* is often the only peer having full access information, we use it as a distributor broadcasting the synchronization messages to all other network peers.

Algorithm 3 shows the steps taken by CAMFLOOP when a vehicle veh receives the synchronization information of

[1] In the protocol WiFi Direct [10] that we use for the simulations presented in this paper, this functionality is provided by the Group Owner, see Sect. IV.

Algorithm 2: *Method:* $synchronize(veh)$

merged buffer $\leftarrow$ *transmission buffer* $\cup$ *opportunity buffer*
if $veh = manager$ **then**
  for all $veh_r \in$ *ad-hoc network* $\setminus\{manager\}$ **do**
    $send\_message(veh, veh_r,$ *merged buffer*$)$.
  end for
else
  $send\_message(veh, manager,$ *merged buffer*$)$.
end if

Algorithm 3: *Method:* $update\_buffers(veh)$

1: $update\_adhoc\_members \leftarrow$ **false**
2: **for all** m_{rb} in $received\_messages(veh)$ **do**
3:   **if** $\exists m_{tb}$ in *transmission buffer* : $m_{tb}[id] = m_{rb}[id]$ **then**
4:     **if** $m_{rb}[tag\_time] \leq leaving\_time$ **and** $m_{rb}[tag\_owner] \neq veh$ **then**
5:       Remove m_{tb} from *transmission buffer*.
6:       Store m_{rb} in *opportunity buffer*.
7:     **end if**
8:   **else if** $\exists m_{ob}$ in *opportunity buffer* : $m_{ob}[id] = m_{rb}[id]$ **then**
9:     **if** $m_{ob}[tag\_owner] = m_{rb}[tag\_owner]$ **then**
10:       $m_{ob}[tag\_time] \leftarrow m_{rb}[tag\_time]$.
11:       **if** $m_{rb}[tag\_time] \geq leaving\_time$ **then**
12:         Remove m_{ob} from *opportunity buffer*.
13:         $m_{rb}[tag\_owner] \leftarrow veh$.
14:         Store m_{rb} in *transmission buffer*.
15:         $update\_adhoc\_members \leftarrow$ **true**.
16:       **end if**
17:     **else**
18:       **if** $m_{ob}[tag\_time] \geq m_{rb}[tag\_time]$ **then**
19:         $m_{ob}[tag\_time] \leftarrow m_{rb}[tag\_time]$.
20:         $m_{ob}[tag\_owner] \leftarrow m_{rb}[tag\_owner]$.
21:         $update\_adhoc\_members \leftarrow$ **true**.
22:       **end if**
23:     **end if**
24:   **else**
25:     **if** $leaving\_time > m_{rb}[tag\_time]$ **then**
26:       Store m_{rb} in *opportunity buffer*.
27:     **else**
28:       $m_{rb}[tag\_time] \leftarrow leaving\_time$.
29:       $m_{rb}[tag\_owner] \leftarrow veh$.
30:       Store m_{rb} in *transmission buffer*.
31:       $update\_adhoc\_members \leftarrow$ **true**.
32:     **end if**
33:   **end if**
34: **end for**
35: **if** $(update\_adhoc\_members =$ **true**$)$ **then**
36:   $synchronize(veh)$.
37: **end if**

one of the other peers. It handles all received messages m_{rb} separately as described by the loop between lines 2 and 34. Further, it checks if a copy of m_{rb} is already in the transmission buffer (lines 3 to 7), in the opportunity buffer (lines 8 to 23), or not yet stored in veh (lines 24 to 33).

If a copy m_{tb} of m_{rb} is already in the transmission buffer, m_{rb} has an earlier tag time than the proposed leaving time of veh, and the tag owners of m_{tb} and m_{rb} are different, the tag owner of m_{rb} will most likely be able to deliver the message earlier than veh. Therefore, m_{rb} is moved from the transmission buffer to the opportunity buffer.

If m_{rb} has a copy m_{ob} in the opportunity buffer of veh, CAMFLOOP checks if m_{rb} and m_{ob} have the same tag owner. If that is the case, the tag time of the message is updated. Moreover, if the adjusted tag time is later than the proposed leaving time of vehicle veh, veh can speed up the delivery of the message. Therefore, m_{rb} is moved into the transmission buffer (lines 9 to 16). If the tag owners of m_{ob} and m_{rb} are different, and the tag time of m_{rb} is earlier than that of m_{ob}, veh learns about a new vehicle that can deliver m_{ob} faster than those, it previously knew about. Therefore, it amends the tag time and owner information of m_{ob} accordingly (lines 17 to 23). If message m_{rb} is new to veh and its tag time is before its estimated leaving time, the message will be stored in the opportunity buffer, otherwise in the transmission buffer (lines 24 to 33).

A greater change than just updating the tag time should be synchronized with the peers in the ad-hoc network. To model that, we use the variable $update\_adhoc\_members$ and the pseudo code between lines 35 and 37.

IV. SIMULATOR

The deterministic nature of our original Context-Aware Message Flooding Protocol simplified the formal verification that the delivery is as fast as possible and keeps the number of delivered messages to a minimum, see [7]. CAMFLOOP, the current version, is reshaped to increase its robustness also in relation to unpredictable behavior by the drivers. Modeling and proving such behavior is more complex and affords the usage of probabilistic formal techniques like PATL [21]. Moreover, we are interested in quantifiable data like the average improvement of the data delivery. Since gaining such data on a real test-bed has not been to our disposal, we opted on using a realistic mobility simulator instead.

As our basis, we selected the versatile fixed-increment time progression simulator *Simulation of Urban MObility (SUMO)* [9], that offers a highly realistic simulation environment. Another advantage of SUMO is a powerful interface that allows us to carry out Python scripts through a Python-TraCI Library whenever a time increment is executed. Using this interface, we created 8,423 lines of Python code to allow SUMO simulating both, the functionality of CAMFLOOP and the underlying ad-hoc network technology WiFi Direct [10]. SUMO also offers an intuitive *Graphical User Interface (GUI)* that facilitates the design of realistic road networks and road user behaviors including the introduction of non-deterministic driving activities by the drivers.

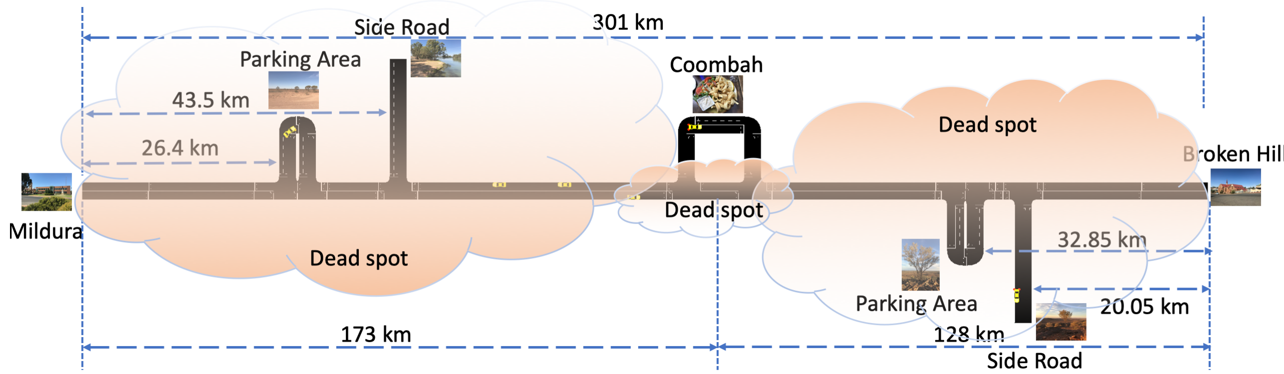


Fig. 1. Simulated Outback scenario.

Since CAMFLOOP is particularly advantageous for large sparsely populated areas with huge dead spots that may span over several hundreds of kilometers, our simulations cover such regions. We decided to model a part of the Silver Highway in the Australian Outback that connects the towns Mildura in Victoria and Broken Hill in New South Wales. The distance between the settlement Wentworth, New South Wales, outside Mildura and Broken Hill is 301 km, and the only location on the road that can provide travellers with network access, is an open WiFi access point at the roadhouse Coombah, 128 kilometers away from Broken Hill.

The aforementioned itinerary is depicted in Fig. 1. Along this road, a few unpaved side roads leave to very remote places deeper in the Outback, which are sometimes hundreds of kilometers away. It is realistic to assume that the connectivity in those settlements is absent such that vehicles directed towards them will not have cellular connectivity for a very long time. In our simulation, we added two side roads located $43.5km$ away from Mildura and $20.05km$ away from to Broken Hill, that traffic in both directions can non-deterministically select to take. Vehicles taking a side road leave our scenario without being able to deliver messages stored by them. In Sect. V, we report about the effect of such "vanishing" vehicles on the overall performance of CAMFLOOP.

Furthermore, we included two parking areas in our simulated scenarios. They are used to simulate short stops of the drivers, that introduce approximately five minutes delay to the previously predicted leaving times of the dead spot and therefore may influence the dynamics of the message exchange and the delivery times of the messages. The parking areas are located $26.4km$ and $32.85km$ away from Mildura and Broken Hill respectively. For Coombah at night and both parking lots at all times of day, the likelihood, that a vehicle makes a stop there, is 15%, while it is 25% for pausing at Coombah in the daytime. We assume that Coombah and the parking lots are sufficiently close to the main road such that ad-hoc networks between parking vehicles and passing traffic can be built up.

To use a realistic traffic volume in our SUMO simulation, we applied a traffic density distribution based on true traffic statistics. In [22], the government of New South Wales offers the hourly number of vehicles passing a point around 70 kilometers north of Wentworth for each direction on certain days in the past. The traffic density varied between two and three vehicles per hour in the night and 38 between 11:00 and noon.

Moreover, we are interested in the influence of the dead spot size on the performance of CAMFLOOP. To make that possible, we modeled not only the full Silver Highway scenario but also versions with artificially reduced distances. In this way, we also created scenarios in which the dead spot sizes are only 10%, 25%, and 50% of the real one, i.e., span only $30.1km$, $75.25km$, and $150.5km$ respectively. We conducted a lot of simulations for all scenarios with varying likelihoods to take a side road and different maximum speeds of the vehicles. A single simulation run over 24 hours for the 10%, 25%, 50%, and full Silver Highway scenarios lasts on average $426s$, $939s$, $1829s$, and $4362s$ respectively using an Intel i5-8259U processor with a frequency of up to 3.8 GHz based on a RAM of 64 GBytes. We discuss the aggregated results of our tests in Sect. V-A.

Standardized Vehicular Ad-Hoc Network (VANET) technologies enabling vehicle-to-vehicle and vehicle-to-infrastructure messaging are *Dedicated short Range Communication (DSRC)* in the USA and *ITS-G5* in Europe [1]. While these technologies will promulgate in the next years, *WiFi Direct* [10] is a protocol for ad-hoc networks, that is already available in most of the current smart phones. It is based upon the IEEE 802.11 infrastructure mode.

Since WiFi Direct can be used for vehicle-to-vehicle communication without spending extra costs in communication equipment, we already used it in our previous work. Unlike in traditional WiFi networks, the particular role, a peer plays in WiFi Direct, is dynamically selected. Therefore, each peer needs to be able to be both, a network manager, called *Group Owner* (*GO*), and a client. Since the GO is the only peer that a-priori has the full access data of the peers, we apply it as the *manager* in CAMFLOOP (see Algo. 2). WiFi Direct uses the social channels 1, 6, or 11 in the 2.4 GHz band for discovery. If two or more peers discover each other, they negotiate through a three-way handshake procedure which

TABLE I
SOME SIMULATION RESULTS OF CAMFLOOP.

Sce-na-rio	Delivery Factor		Delivering vehicles			Improvement	
	Flood.	Our prot.	Originator None	Originator Conn.	Ano.	Ano. only	All
10%	17.25	1.359	27.10	23.23	49.23	57.09	28.16
25%	28.14	1.518	14.47	24.00	61.20	69.90	36.70
50%	49.22	1.585	8.87	23.19	67.64	60.66	41.08
Full	93.57	1.688	5.17	19.59	74.99	59.46	44.61

station will be GO, and the GO executes the Dynamic Host Configuration Protocol (DHPC) to provide the clients with unique IP addresses.

In previous work, we conducted field tests with Android smart phones and tablets connecting each other using WiFi Direct. Amongst others, we surveyed the distributions of the times needed to discover other devices, to connect with each other, and to exchange data in WiFi Direct, see [5]. The average times were $1.81s$ for the discovery, $2.29s$ for the connection including the GO selection, and $0.97s$ to exchange the system statuses of all peers. We utilized these distributions to simulate the ad-hoc network behavior in SUMO and conducted a test series with varying network ranges of WiFi Direct which is described in Sect. V-B. Our previous tests showed that WiFi Direct often exceeds the official maximum range of $200m$, since connections over $250m$ could be frequently realized [5].

V. SIMULATION RESULTS

Using SUMO, we conducted numerous simulation runs to analyze various aspects of CAMFLOOP. In particular, we were interested in the following questions:

- Is CAMFLOOP robust enough to be used in practice, or do we need further improvements of our protocol?
- Is WiFi Direct a sufficient basis for building the ad-hoc networks of CAMFLOOP, or do we have to rely on more capable VANET protocols?

Answers to these questions are given in the two following subsections.

A. Scenario-based results

Below, we discuss quantitative performance results of CAMFLOOP depending on varying input parameters like different dead spot sizes, maximum speeds of the vehicles, ranges of WiFi Direct, and likelihoods that a vehicle surprisingly takes one of the side roads. Table I depicts a number of properties for each of the four dead spot sizes, we have simulated with SUMO. We show the averages of 20 simulation runs, each covering 24 hours. In all these runs, we set the maximum speed of the vehicles to $110km/h$ which is also the effective maximum speed on most parts of the Silver Highway. Further, we used the official maximum range of $200m$ for WiFi Direct, Finally, we selected a likelihood of 0.125%, that a vehicle[2] takes a side road in contrary to the itinerary selected in its route guidance system.

[2]While we have no data, assuming that one of 800 vehicles uses a side road without telling it to its route guidance system, seems sensible.

At first, we compare the average numbers of duplicates produced by CAMFLOOP with those of a traditional flooding protocol, in which every bearer of a message delivers it via the cellular network after having left the dead spot. The statistics for the flooding protocol are listed in the second column of Tab. I and those for CAMFLOOP in the third one. For the flooding protocol, the number of duplicates grows exponentially over the dead spot sizes. This results from the fact that, in larger dead spots, a vehicle takes part in more ad-hoc networks, such that a message is distributed to a greater number of peers. In contrast, the increase of the duplicates in CAMFLOOP is only linear indicating that it is better suited for more extensive dead spot sizes than the flooding protocol. Altogether, the numbers of duplicates produced by CAMFLOOP are smaller by factors ranging from 12.7 for the 10% scenario to 55.4 for the simulation of the full distance between Mildura and Broken Hill, than produced with the flooding protocol.

At next, we check the likelihood that messages are delivered by the vehicles in which they are created or by other ones. The fourth column of Tab. I describes the percentages that the message creator delivers a message since it could not connect with any other vehicle between creating the message and leaving the dead spot. Also the fifth column refers to message deliveries by the creator, but in this case after having been a member in at least one ad-hoc network. The sixth column shows the percentage of messages[3] delivered by a vehicle other than their creators. We consider it interesting that the share of other vehicles than the originator delivering the messages exceeds the one of the message creator significantly, especially in the larger dead spots. Looking deeper into the simulation logs suggests two reasons for this somehow surprising effect: One is the simulated roadhouse Coombah with its WiFi accessibility. It is helpful for vehicles not stopping there themselves or creating messages shortly after having passed it since the messages may be handed over to other traffic taking a break at Coombah. The other effect results from overtaking traffic which can happen since SUMO realistically models quite varying average speeds of the different vehicles. That causes vehicles overtaking each other, through which the slower traffic may hand over its messages to the faster one for an earlier delivery.

In the last two rows of Tab. I, we depict the average improvement of delivery time by CAMFLOOP in percent (e.g., if a vehicle creates a message $60min$ before leaving the dead spot but can hand it to another one that regains cellular network connectivity already after $45min$, i.e, $15min$ earlier that the creator, we apply an improvement of 25%). The seventh column describes the average of these gains for messages not delivered by their creators, while the last column depicts the average improvement over all messages. The results show that the improvement in delivery time is more pronounced in the larger dead spots. For instance, the mean improvement of a message created in the 10% scenario is

[3]The difference to 100% results from messages that could not be delivered at all. We discuss this aspect later in this subsection.

TABLE II
EVALUATING WEAKNESS OF CAMFLOOP.

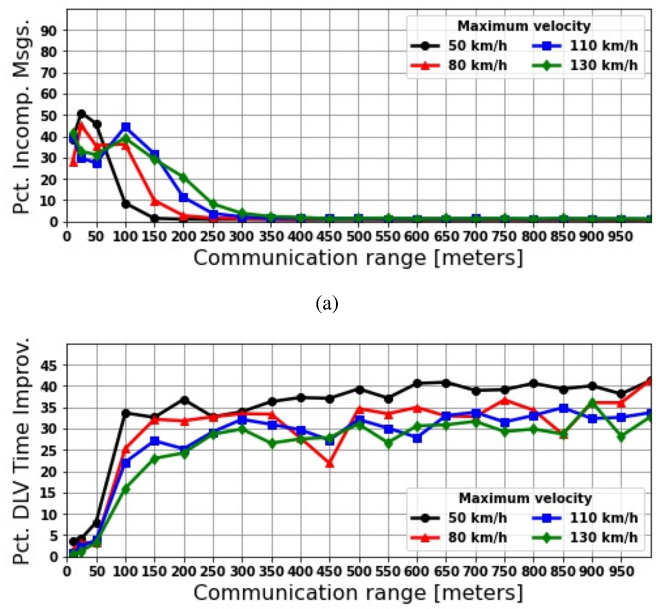
Sce-na-rio	Pct. side road	Gen. mes-sages	Not delivered messages Avg. no.	Orig. (could have sent)	Other (could have sent)	Not best deliv.
10%	0.125	150	0.130	0.067	0.033	2.13
	1	143	0.804	0.245	0.280	2.46
	4	141	3.014	1.064	0.496	2,32
25%	0.125	312	0.080	0.080	0.000	1.91
	1	312	0.769	0.401	0.208	2.16
	4	299	2.993	1.455	0.736	2.33
50%	0.125	584	0.103	0.051	0.026	2.20
	1	571	0.569	0.306	0.140	2.23
	4	569	2.425	1.503	0.554	2.26
Full	0.125	1164	0.048	0.031	0.015	2.64
	1	1135	0.634	0.441	0.159	2.39
	4	1106	3.029	1.997	0.715	1.94

28.16% which corresponds to an earlier delivery by $2.4min$ on average assuming vehicles travelling with $110km/h$. In contrast, the gain in the real dead spot between Mildura and Broken Hill is 44.61% or $36.6min$ on average, which can be very helpful particularly in emergency situations.

In order to further increase the robustness of our protocol, we analyzed and evaluated its performance in a highly unpredictable environment. The results of these tests are depicted in Tab. II. Since some effects depend on the likelihoods that a vehicle takes a side road, we show the results not only for a likelihood of 0.125% that a vehicle enters a side road contrary to the setting of its route guidance system but also for values of 1% and 4% respectively. In the third column of Tab. II, we also present the average numbers of messages generated in the 20 runs of 24 hours each since we had no space for that in Tab. I.

The fourth, fifth, and sixth columns describe the likelihoods that messages are not delivered due to vehicles taking the side roads. In the fourth column we list the percentages of messages that were not delivered at all. The values range from between around 0.1% for a likelihood of 0.125% and more than 3% if the likelihood is 4%. Several of the failed deliveries, however, occur when a vehicle carrying a message does not join an ad-hoc network with other vehicles before entering the side road. No protocol can mitigate this situation.

The most problematic case for CAMFLOOP is that the originator of a message leaves the dead spot normally at one of its borders or stops in Coombah, but does not deliver the message there. That can happen if the message creator thinks that another vehicle already sent the message, but the selected vehicle took a side road instead. In this case, our protocol prevents the delivery of a message that would have been sent via the cellular network, otherwise. Such message losses happen in less than 0.1% of all cases if we assume a likelihood of 0.125% for sudden deviations, but with a probability of one to two percent if the likelihood is 4%. Therefore, we might consider to deliver messages always by their originators after leaving the dead spot, even if a faster vehicle delivers them earlier. This avoids the above mentioned weakness by CAMFLOOP, albeit at the cost of creating additional duplicates. For instance, in the full



(a)

(b)

Fig. 2. Evaluation of CAMFLOOP with respect to (a) Failed message exchanges within ad-hoc networks, (b) Improvement of the delivery time.

scenario, the number of duplicates would increase from 1.688 to 2.438. Nevertheless, depending on the reliability demands on CAMFLOOP, this slight increase might be tolerable.

The sixth column describes cases in which the originator of a message entered a side road but handed the message over to another vehicle in an ad-hoc network that leaves the dead spot. This vehicle, however, does not send the message since it believes that the original vehicle can deliver it faster. Here, a flooding protocol would avoid this error but the low probabilities of less than 0.04% with the likelihood of 0.125% to take a side road and less than 0.8% with 4% will outweigh the drastically increased numbers of duplicates.

The last column shows the number of messages that were delivered but not in the best possible time. This can happen when vehicles are significantly slower or faster than previously estimated. In our tests, around 2% of all messages are delivered in a sub-optimal time. Since the number of dramatically delayed deliveries of more than a few minutes was negligible, we consider this result acceptable.

B. Influence of WiFi Direct Variants

To find out if WiFi Direct [10] is a suitable short range communication technology for the ad-hoc network creation in CAMFLOOP, in particular in the case of opposing traffic, that is passed within a few seconds, we ran several simulations of the 10% dead spot scenario in which we tested varying ranges of WiFi Direct for each of four different maximum speeds of the simulated vehicles. The results of these tests are illustrated in Fig. 2. The curves of Fig. 2(a) depict the share of failed ad-hoc network connections depending on the network ranges and maximum speeds. They promulgate that, with speeds of up to $80km/h$, a range of $200m$ is sufficient to create ad-hoc network connections reliably, while we need

$250m$ for $110km/h$ and $300m$ for $130km/h$. Thus, WiFi Direct with its up to $250m$ maximum range is at the edge at least for the faster maximum speeds tested.

Being at the edge, however, means that still many ad-hoc networks might be successfully built. That is particularly true as CAMFLOOP is quite robust against break downs during a session. Our simulations showed that, even with a communication range of less than $250m$, not more than 1% of failed message exchanges in ad-hoc networks lead to a loss in relevant context sharing. To understand the practical impact of these aspects better, we also checked the improvement of message deliveries by CAMFLOOP (see also Tab. I) for the different combinations of maximum network ranges and speeds. The corresponding results are shown in Fig. 2(b). We can see that CAMFLOOP provides a significant improvement for network ranges starting with $100m$ with just slight increases for those with longer ranges. The effectiveness of our protocol will be only slightly reduced by around 5% for the typical ranges of WiFi Direct between $200m$ and $250m$. This tempts us to assume that, as long as VANETs with higher maximum ranges are not common, WiFi Direct is a good fit to be used with CAMFLOOP, also when the maximum speeds of the vehicles are beyond $100km/h$.

VI. CONCLUSION

We introduced the updated version CAMFLOOP of our Context-Aware Message Flooding Protocol for the mitigation of dead spots between vehicles and the infrastructure. In contrast to the original variant [7], CAMFLOOP handles nondeterministic drivers' behavior like making breaks or changing the initial route. We evaluated CAMFLOOP by carrying out simulations based on the traffic simulator SUMO [9]. As long as the demanded reliability parameters are not too high, the results of our tests are promising, and the current version of CAMFLOOP seems robust enough even on uncertain conditions for its use in practice. As a precaution, however, one should consider to let the creator of a message deliver it after regaining cellular network connectivity, even if that increases the number of produced duplicates slightly.

In spite of the overall good results, we study variants of the protocol in which the vehicles also send messages from their opportunity buffers. The likelihood to do that, might thereby depend on the freshness of the messages. Moreover, we have also experimented with different traffic volumes as well as varying numbers and placements of stops providing WiFi like Coombah. Due to the space limit, we will report on the results of these tests in future work. Finally, we plan to develop a formal description of CAMFLOOP using a probabilistic logic like PATL [21] and to verify relevant properties based on these models.

REFERENCES

[1] J. E. Siegel, D. C. Erb, and S. E. Sarma, "A Survey of the Connected Vehicle Landscape — Architectures, Enabling Technologies, Applications, and Development Areas," *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[2] NPerf, "3G / 4G / 5G Coverage Map," https://www.nperf.com/, 2020, accessed: 2021-04-18.

[3] C. F. Mecklenbräuker, A. F. Molisch, J. Karedal, F. Tufvesson, A. Paier, L. Bernado, T. Zemen, O. Klemp, and N. Czink, "Vehicular Channel Characterization and Its Implications for Wireless System Design and Performance," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1189–1212, 2011.

[4] P. E. Driesen, "Prediction of Multipath Delay Profiles in Mountainous Terrain," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 3, pp. 336–346, 2000.

[5] E. Puka and P. Herrmann, "Data Dissemination for Vehicles in Temporary Cellular Network Dead Spots," *International Journal of Cyber-Physical Systems (IJCPS)*, vol. 1, no. 2, pp. 38–55, 2019.

[6] J. A. E. Meyer, E. Puka, and P. Herrmann, "Utilizing Connectivity Maps to Accelerate V2I Communication in Cellular Network Dead Spots," in *6th International Conference on Internet of Vehicles (IOV)*, no. LNCS 11894. Kaohsiung, Taiwan: Springer-Verlag, 2019, pp. 76–87.

[7] E. Puka, P. Herrmann, and A. Taherkordi, "Hybrid Context-aware Message Flooding for Dead Spot Mitigation in V2I Communication," in *92nd IEEE Vehicular Technology Conference (VTC-Fall)*. Victoria, BC, Canada (Virtual): IEEE VTS, 2020, pp. 1–7.

[8] F. F. Tseng, D. P. Filev, and I. H. Makki, "Vehicular Connectivity Map," U.S. Patent 9 775 128 B2, Sept. 26, 2017, U.S. Patent 9 775 128 B2.

[9] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic Traffic Simulation using SUMO," in *IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2018, pp. 2575–2582.

[10] Wi-Fi Alliance, P2P Technical Group, "Wi-Fi Peer-to-Peer (P2P) Technical Specification v1.7," 2016.

[11] H.-Y. Hsieh and R. Sivakumar, "On Using the Ad-hoc Network Model in Cellular Packet Data Networks," in *3rd ACM International Symposium on Mobile Ad hoc Networking & Computing (MobiHoc)*. ACM, 2002, pp. 36–47.

[12] Y. Lmoumen, Y. Ruichek, and R. Touahni, "Multi-Hop Communications inside Cellular Networks: A Survey and Analysis," *International Journal of Electrical and Electronic Engineering & Telecommunications*, vol. 8, no. 6, Nov. 2019.

[13] C. Huang, K. Lan, and C. Tsai, "A Survey of Opportunistic Networks," in *22nd International Conference on Advanced Information Networking and Applications - Workshops (AINA Workshops)*, Ginowan, Japan, 2008, pp. 1672–1677.

[14] M. J. Khabbaz, C. M. Assi, and W. F. Fawaz, "Disruption-Tolerant Networking: A Comprehensive Survey on Recent Developments and Persisting Challenges," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 607–640, 2012.

[15] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad hoc Networks," *IEEE Communications Magazine*, vol. 44, no. 11, pp. 134–141, 2006.

[16] M. Alajeely, R. Doss, and A. Ahmad, "Routing Protocols in Opportunistic Networks —- A Survey," *IETE Technical Review*, vol. 35, no. 4, pp. 369–387, 2018.

[17] F. J. Martinez, C. K. Toh, J.-C. Cano, C. T. Calafate, and P. Manzoni, "A Survey and Comparative Study of Simulators for Vehicular Ad hoc Networks (VANETs)," *Wireless Communication Mobility Computing*, vol. 11, pp. 813–828, 2011.

[18] B. Liu, D. Jia, J. Wang, K. Lu, and L. Wu, "Cloud-Assisted Safety Message Dissemination in VANET–Cellular Heterogeneous Wireless Network," *IEEE Systems Journal*, vol. 11, no. 1, pp. 128–139, Mar. 2017.

[19] C. Olaverri-Monreal, J. Errea-Moreno, and A. Díaz-Álvarez, "Implementation and Evaluation of a Traffic Light Assistance System Based on V2I Communication in a Simulation Framework," *Journal of Advanced Transportation*, p. 11 pages, 2018.

[20] L. C. Bento, R. Parafita, and U. Nunes, "Intelligent Traffic Management at Intersections Supported by V2V and V2I Communications," in *15th International IEEE Conference on Intelligent Transportation Systems*, Anchorage, AK, USA, 2012, pp. 1495–1502.

[21] T. Chen and J. Lu, "Probabilistic Alternating-time Temporal Logic and Model Checking Algorithm," in *4th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*. Haikou, China: IEEE, 2007, pp. 35–39.

[22] NSW Government, "Traffic Volume Viewer," www.rms.nsw.gov.au/about/corporate-publications/statistics/traffic-volumes, 2021, accessed: 2021-04-18.