Utilizing Connectivity Maps to Accelerate V2I Communication in Cellular Network Dead Spots

Jon Arild Ekberg Meyer[0000−0002−7725−2084], Ergys Puka[0000−0002−4257−3232],
and Peter Herrmann[0000−0003−3830−466X]

Norwegian University of Science and Technology (NTNU), Trondheim, Norway
jon.ae.meyer@gmail.com, ergys.puka@ntnu.no, herrmann@ntnu.no
http://www.ntnu.no/iik

Abstract. On many roads in rural and mountainous areas, the cellular network connectivity is intermittent and dead spots, i.e., zones without any coverage, are frequent. In previous work, we developed a data dissemination protocol to accelerate the transmission of messages in dead spots. It combines the cellular network with short-living ad-hoc networks between vehicles. A car in a dead spot can forward messages directed towards the environment, to the peer in its ad-hoc network that will leave the dead spot first, effectively reducing the delay. An issue, however, is to reliably identify the peer that is most likely the first one regaining cellular network coverage. This problem can be solved if the borders of the dead spot, the vehicles are in, are previously known. For that, we use a novel technology named dead spot prediction. Here, vehicles conduct local connectivity measurements that are aggregated to so-called connectivity maps describing the locations of dead spots on a road system. In this article, we introduce the combination of the data dissemination protocol with dead spot prediction. Particularly, our protocol is amended such that connectivity maps are considered when deciding which vehicle leaves a dead spot first. Since currently only few publicly available works about dead spot prediction exist, we further created a prototype of such a predictor ourselves that will be discussed as well.

Keywords: Cellular Network Access · Dead Spots · Data Dissemination Protocol · Ad-hoc Network · Dead Spot Prediction · Connectivity Map.

1 Introduction

Intelligent Transport Systems (ITS) in the automotive sector rely on network connectivity. The so-called vehicle-to-infrastructure (V2I) communication is usually carried out using cellular networks [17]. Thanks to the emerging 5G technology, even the vehicle-to-vehicle (V2V) communication, i.e., interactions between cars, will be partially handled by cellular networks as well [13].

A problem of communication based on cellular networks, however, is the varying network coverage. In real life, we regularly come across dead spots, i.e., areas without sufficient cellular network connectivity. Dead spots can be particularly found in sparsely populated areas since the cell tower infrastructure is often driven by the number of people living in an area [10]. Also mountainous terrain makes the network coverage frail as mountains and hills tend to cause echoes deteriorating the radio reception [4]. Own tests showed that the size of dead spots in very remote areas like the Australian Outback can be really large and extend hundreds of kilometers (see [14]).

To mitigate the effect of dead spots, we developed a data dissemination protocol for the transport of data from vehicles to the fixed infrastructure [14]. When vehicles have no cellular network access, they build ephemeral ad-hoc networks with other cars in their area. If one of the peers in such an ad-hoc network also has cellular network access, it can relay the messages of the other network members. If all peers are in the dead spot, the messages are forwarded to the vehicle that most likely regains cellular network connectivity first. Thus, it can send the messages earlier than its peers, and the overall transmission process is accelerated. We developed a prototypical application based on WiFi Direct [22].

A problem to be solved by the data dissemination protocol, is to find out which vehicle in an ad-hoc network has the highest probability to leave the dead spot first. In the original version, we select the car that lost cellular network access first since, assuming similar average speeds, it is supposed to be also the first one having crossed the dead spot [14]. This approach is easy to realize since we only need local connectivity measurements in the vehicles. However, it tends to be coarse since vehicles may have different speeds and can take varying routes. To overcome this weakness, we combine the data dissemination protocol with dead spot prediction. In this approach, so-called connectivity maps [20] describing dead spot areas are used. These maps inform the vehicles about the cellular network connectivity on their way. Then, they can adapt their communication accordingly (see [2, 8, 21]).

Since we could not find any dead spot predictors and connectivity map generators, we created our own prototype [11]. In this article, we sketch this development and discuss that one can infer from the testing results that dead spot prediction will be scalable. Moreover, we present the amendment of the data dissemination protocol such that the decision, which car in an ad-hoc vehicular network shall relay the messages, indeed incorporates aggregated dead spot prediction information.

The paper is organized as follows: In Sect. 2, we report on patents for connectivity prediction systems and some other related work. Thereafter, we sketch the prototype for dead spot prediction and discuss results of experimenting with it in Sect. 3. In Sect. 4, we introduce the data dissemination protocol. The extension of the protocol is described in Sect. 5 followed by a conclusion.

2 Related Work

Several patents provide evidence that the automotive industry has significant interest in dead spot prediction. In [8], Bosch has a distributed architecture patented in which vehicles compute dead spot elongations locally by conducting connectivity measurements. Then they transmit the local data to a central server that forwards it to other vehicles. Using the dead spot elongation data from other cars, a vehicle can decide if it needs to start a dead spot mitigation strategy. That is necessary to handle wireless applications that will not be completed when reaching the next dead spot. A similar architecture is patented by Ford [21]. The authors, however, sketch only shortly that they use a remote server but concentrate on the system layout in the vehicles. In particular, they define the structure of the dead spot prediction information that is realized by special connectivity maps [20]. In contrast, IBM does not mention a central server in its patent [2]. Instead, a mobile device takes current and historic wireless service data as well as general information about the environment (e.g., the presence of tunnels) into consideration. Using intelligent learning systems, these data are analyzed and aggregated to a predictive model anticipating dead spots.

Also dead spot mitigation strategies are protected. Bosch [16] and IBM [2] patented improvements for streaming services used in cars. Before reaching a dead spot, additional streaming data is transmitted and locally stored in the vehicle. This extra data is played while the vehicle passes the dead spot. Further, IBM has a way to ease the interruption of phone calls in dead spots patented [18]. The user is notified about the reason and the duration of the interruption, and the phone call is automatically reconnected after leaving the dead spot.

Another aspect of our work is about using hybrid systems combining cellular and vehicular networks. They are applied for various purposes (see, e.g., [12]) but, except for our own work, we only found one other approach utilizing them for dead spot mitigation. Eltahir et al. [5] use vehicles as relay stations such that a car in a dead spot may transmit messages via a number of other vehicles until reaching one that has cellular network access. In contrast to our approach [14], however, they neither allow to store data in vehicles nor to utilize their directions and positions in a dead spot. Therefore this approach can mitigate only smaller dead spots that contain sufficient traffic. For instance, in the Australian Outback where traffic is low and the dead spot size huge, this approach would not work.

3 Building Connectivity Maps

Our own prototypical dead spot predictor [11] uses an architecture close to the one patented by Bosch [8]. It extends, however, the functionality of the central server that not only forwards data received by the vehicles but also aggregates them to a connectivity map. The vehicles are provided with excerpts of this map containing dead spot predictions relevant to them. In this way, a vehicle does not need to aggregate the data from several other cars itself.

Local connectivity may change over time, for instance, due to network congestion, differing network use, or a breakdown in the cellular infrastructure (see [15]). The central server application therefore needs to constantly process incoming connectivity data. Further, newer data should be weighted higher than older one to keep the connectivity maps up-to-date.

Our prototype uses Android devices in the vehicles to collect cellular network statistics but one can also use other connectivity sensor techniques as long

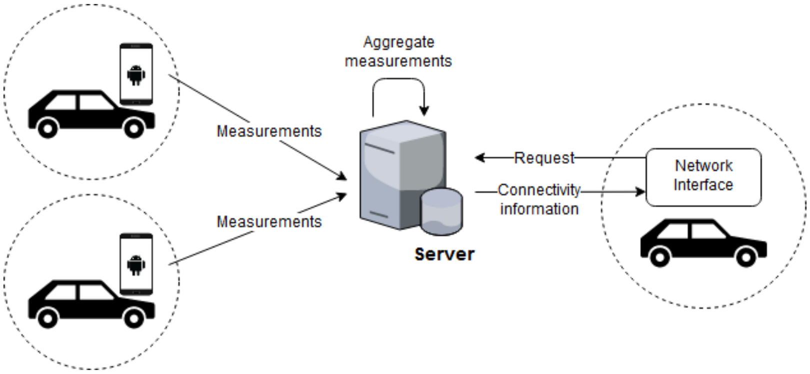


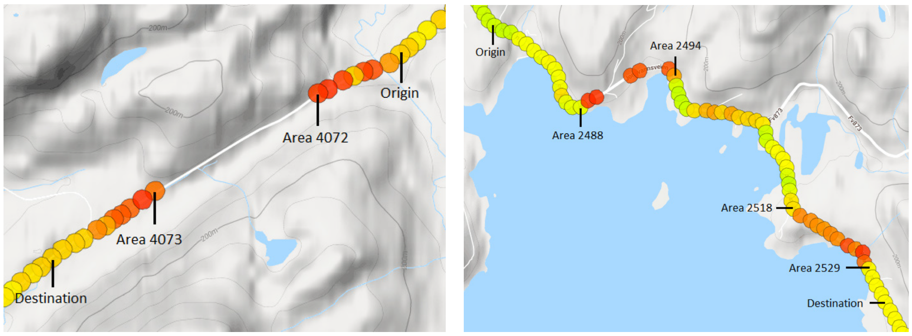
Fig. 1. The architecture of the dead spot prediction prototype.

as the transmitted data follows the expected format. The sensed connectivity parameters include round-trip time, signal strength, jitter ratio, and packet loss. They can easily be extended in future versions if necessary. Each data point also includes the current GPS position where the measurement was taken. After receiving this information from various sources, the server continuously aggregates it into connectivity maps. The vehicles can then request information about upcoming dead spots from the server. This is shown in Fig. 1.

All client measurements taken inside a geographical area with a diameter of 50 meters are conjoined. The central server aggregates the corresponding data points to a connectivity value for each geographical area. It applies special functions that reflect the particular needs of the applications for which a cellular network connection is used. For instance, the jitter ratio is more important when videos are streamed than when a text document is downloaded. Since a connectivity map is composed of the connectivity values in the geographical areas, the server produces separate maps for the different aggregation functions.

To receive a connectivity map, a vehicle provides the central server with its planned route, e.g., from the route guidance system. In our prototype, we use the format of the Google Directions API [6]. Moreover, the vehicle selects the desired aggregation function. The server then sends the connectivity map excerpt as a list of geographical area markers (including their coordinates), each referring to an area where a dead spot starts resp. ends, as well as the dead spot lengths. Two such excerpts are depicted in Fig. 2 where colored dots describe the connectivity values of the geographical areas. The greenish color of a dot refers to good connectivity in the represented area, while yellowish shows intermediate and reddish bad coverage. Missing dots indicate tunnels in which no GPS measurements can be taken. The excerpts reveal the following dead spots:

- On the route depicted in Fig. 2a, a dead spot can be found between the geographical areas 4072 and 4073 which are 977 meter apart. The tunnel in this region is probably the reason for the dead spot.



(a) (b)

Fig. 2. Excerpts of a connectivity map from the Lofoten region in Norway.

- On the route shown in Fig. 2b, there is a dead spot between areas 2488 and 2494. It's length is 661 meters. Also here, tunnels seem to be the reason.
- Another dead spot on the route in Fig. 2b is between the areas 2518 and 2519. It has a length of 642 meters.

In our tests, we merged all measurements for geographical areas with an extent of 50 meters. Moreover, we took a sample every 250 ms such that a car running with 80 km/h produces nine data points in an area. Using this procedure, we collected more than 130,000 samples from two different locations in Norway over a couple of months. About 100,000 points were collected in the rural and highly mountainous location Lofoten, e.g., those shown in Fig. 2. The rest of the data points came from Trondheim, Norway's third largest city. We further added 3,334 data points collected in the context of earlier work about cellular network connectivity in the surroundings of Trondheim [15]. The size needed to store the data is about 30 Mb, which in the modern database world is tiny.

The 130,000 data points can be aggregated on a standard personal computer within a second. Here, the efficiency is ensured using a spatial database extender in the underlying database management system with support for geographical objects. This tool makes it possible to determine geographical proximity for thousands of locations in few milliseconds [11]. Since the application has to pair the GPS coordinates of each measurement point with an area in the database, the overall computation costs are significantly reduced.

However, only three different vehicles were used to produce this amount of data. In a real-world, large scale environment, millions of data points would be produced every day. To get a better understanding if the dead spot prediction approach is scalable, we projected our findings to the overall road traffic in Norway. The results of this analysis will be discussed in the following.

In 2018, motor vehicles in Norway ran around 46 billion kilometers [19]. We could not find information about the average speed of these cars, but detected similar data from the United Kingdom [3]. The average free flow speed of all cars there was 35 mph in 2012 which corresponds to 56.3 km/h. Since the share of motorways in Norway is only around half as large as in the UK and there are more mountainous roads, the average speed will most likely be lower. Thus, we assume an average speed value of 50 km/h: Then the motor vehicles in Norway ran 920 million hours in 2018. Using a sample rate of 250 ms, they could produce up to 13.248 trillion data points in that year. If a data center stores all these samples, e.g., to be able to consider long-term connectivity changes, and a data point is represented by 256 bytes, the overall storage size needed is around 3.4 petabytes. A larger data center should be able to handle this.

In contrast to the solution suggested by Bosch [8], we aggregate the data for geographical areas with an elongation of 50 meters which will save memory. Since the road system in Norway covers 93,870 km [7], one has to keep at most 1,877,400 areas. Using a kilobyte to represent the data of one area, the overall storage size will be just around 1.9 gigabytes.

The price for this more concise way to store the data, however, is that we need to aggregate incoming data points into the connectivity values of the areas. This causes more processing effort than just storing the data. Projecting our experience, that we can aggregate 130,000 data points within a second, to the 13.248 trillion data points, that the cars in Norway could have theoretically produced in 2018, the overall computation time for aggregations will be 102 million seconds on a single PC. That is around 3.2 years such that, according to [1], the distribution of the process on 11 or 12 parallel running computers should be sufficient.

Finally, we have to look on the data transmission. To send 3.4 petabytes from the vehicles to the central server in the course of a year, affords a data transmission rate of around 820 GBit/s for the whole country. According to the fact that 5G will offer 10 GBit/s for a single connection, we consider this amount as doable. The use of compression mechanisms will alleviate the data transmission further.

Altogether, even considering the case that all cars in Norway participate in the dead spot prediction whenever they run, the approach seems scalable.

4 Data Dissemination Protocol

To reduce the waiting time for message transmissions between vehicles and their fixed infrastructure in areas where the cellular network coverage is weak, we developed a special data dissemination protocol [14]. In addition to the communication between vehicles and the infrastructure (V2I) via a cellular network, it uses short-lived ad-hoc vehicular networks (VANET) between close-by vehicles. Applying the ad-hoc network in areas in which the cellular network connectivity is low, a message can be forwarded to a vehicle which is either out of the dead spot or will likely leave it earlier than the message initiator such that the delivery of the message can be expedited.

The prototype, also introduced in [14], uses Android devices in the vehicles that support WiFi Direct technology [22]. In particular, the WiFi Peer-to-Peer

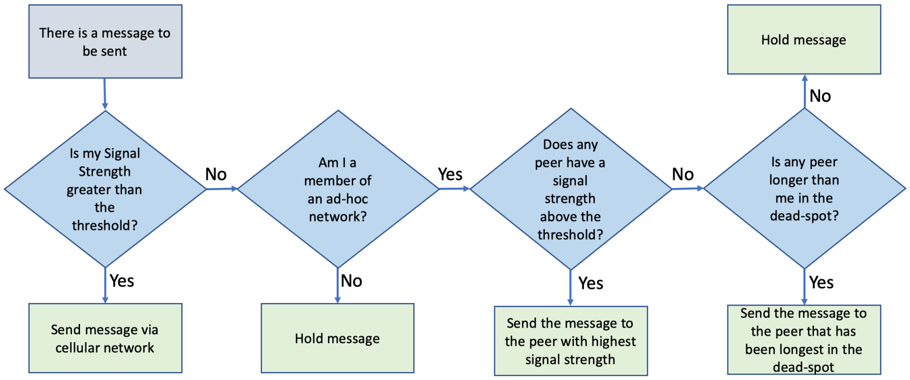


Fig. 3. Steps executed by the data dissemination protocol.

framework (WiFiP2P) was applied to implement WiFi Direct (see also [9]). Besides the fact that Android devices are very common, they allow us to exploit the immense capabilities of the Android OS. For instance, convenient methods to access signal strength measurements and local IP addresses are offered. As discussed below, these are features important for the realization of the protocol.

In the prototype, we use the signal strength to evaluate the cellular network coverage. That can, however, be easily extended to other forms of measurements, e.g., by applying the aggregation functions discussed in Sect. 3. If its current signal strength is below a certain threshold indicating the proximity of a dead spot, a vehicle tries to connect with other ones in its vicinity to form an ad-hoc network. If such a network can be established, the peers in it exchange their IP addresses, current signal strengths, and, if they lost cellular network coverage, the points in time, when that happened. This data is locally stored at the peers and utilized in the various steps of our protocol that are depicted in Fig. 3.

In the first protocol step, a peer veh_s which initiates the transmission of a message to the fixed environment, checks if its mobile network coverage is sufficient. If that is the case, the message is directly sent via the cellular network. Otherwise, veh_s tests if it is already part of an ad-hoc network. If it is not, it simply holds the message and the procedure is executed again when the peer either joins an ad-hoc network or leaves the dead spot.

If veh_s already belongs to an ad-hoc network while it is in a dead spot, it checks by using the locally stored signal strength information of its peers, if any of those has sufficient cellular network coverage. If this is the case, veh_s sends the message to the peer veh_b which has the best signal strength value. Then, veh_b forwards the message immediately via the cellular network.

If no peer has coverage, the message is sent to the one which presumably will leave the dead spot first in order to achieve a message delivery at the earliest opportunity. In the solution presented in [14], veh_s compares the points of time, the peers entered the dead spot, and transmits its message to vehicle veh_l that is already longest in it. As discussed in Sect. 5, we assume that all vehicles have around the same speed on the mostly small and mountainous roads where many dead spots occur. In consequence, veh_l, which entered the dead spot first, will likely be also the first one leaving it. If veh_s itself has been longer in the dead spot than all the other members of the ad-hoc network, it holds the message since it will probably be the first one regaining connectivity.

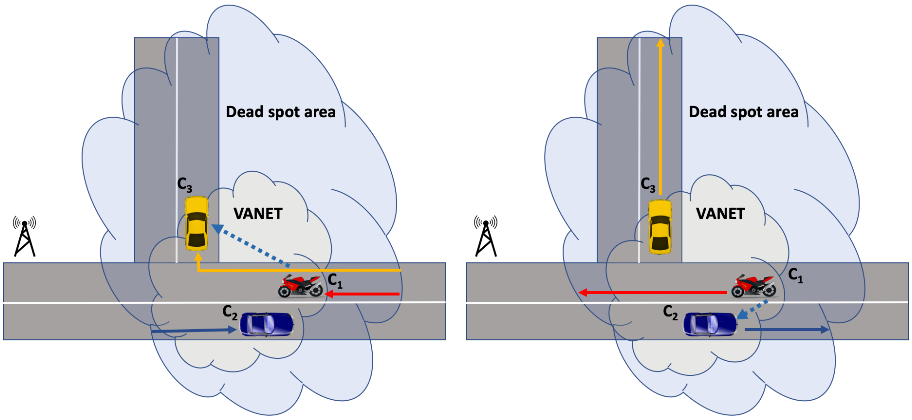
A special case is a vehicle stopping in the dead spot. In this case, it informs its peers and transfers the stored messages to other vehicles. If the stopping vehicle is not a partner in an ad-hoc network, it tries to establish a new one as long as it still stores messages in order to pass them to a peer still moving.

In [14], we also discuss the evaluation of the protocol implementation on Android devices using WiFiP2P. The most time-critical scenario for message handovers between cars is if they run in opposite directions. Since the range of WiFiDirect-based networks is around 200 meters, building an ad-hoc connection, carrying out the protocol, and transmitting messages have to be completed in 6.5 seconds if both vehicles have a speed of 110 km/h. If the cars run with 80 km/h each, the data exchange has to be accomplished in nine seconds while it is 14.4 seconds for vehicles with 50 km/h. According to our test cases, the likelihood to complete the building of the ad-hoc network and the message handover in 6.5 seconds is 71%, while 97% of all trials were successfully carried out in nine seconds and all tests within 14.4 seconds. So, our example implementation seems to be reliable for cars with a speed of up to 80 km/h. But also for a speed of 110 km/h, it seems solid since, when an exchange with another vehicle fails, the sender can retry it with the next one. The likelihood that a message handover is successful in one of the first three trials is 98% when all participating vehicles have a speed of 110 km/h.

5 Updating the Data Dissemination Protocol

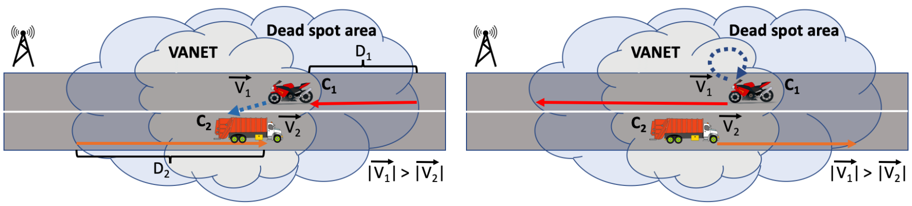
In our original approach [14], the decision to transmit data to the vehicle that has entered the dead spot first, rests on two assumptions: The first one is that the size of a dead spot is about the same for all vehicles. Here, we suppose that the cars use the same cellular network technology and that the network operators apply similar connectivity optimization strategies such that the extents of the dead spots are alike. The second assumption states that all vehicles have similar average speeds while passing it. This is based on the fact that many roads in rural and mountainous terrain, where dead spots are most prevalent, are minor. On this kind of roads, the speeds of the vehicles are often limited by the road quality and not their own driving characteristics. For two reasons, however, this assumption is often imprecise:

1. There can be road crossings in a dead spot. Then, we compare vehicles that possibly take different routes such that the times, they are in a dead spot, can heavily vary. For instance, the red motorcycle in Fig. 4a forwards its messages to the yellow car since that is already longer in the dead spot than its peers.



(a) Original protocol (b) Updated protocol

Fig. 4. Behavior of the data dissemination protocol in dead spots with intersections.



(a) Original protocol (b) Updated protocol

Fig. 5. Behavior of the data dissemination protocol in dead spots with vehicles running at different speeds.

Due to its turn, however, the yellow car has to cover a greater distance in the dead spot than the blue one which regains connectivity earlier.

2. Roads are usually used by different types of motor vehicles which may run with significant speed differences, e.g., when heavy trucks have to overcome steep climbs. This is depicted in Fig. 5a, where the motorcycle reaches the end of the dead spot earlier than the truck since the latter one is very slow. Nevertheless, since the truck is longer in the dead spot, the motorcycle falsely passes its messages to it.

Applying connectivity maps can alleviate these problems since that allows us to predict the time points, at which the vehicles reach the dead spot borders, more precisely. To get such a time estimate, we use connectivity map excerpts for computing both, the distance to the end of the dead spot and the predicted average speed.

To calculate the distance to the dead spot boundary, the current GPS location of the vehicle and the projected route are taken from the route guidance system, e.g., in the format given by the Google Directions API [6]. Further, the end point of the dead spot is retrieved from the connectivity map excerpt and the length of the way to this point is computed.

Determining the average speed is more subtle since it depends on several structural factors. One solution is to apply a pre-defined general average speed value for the vehicle class and the type of road used. While this proceeding is still relatively coarse, we can at least distinguish different types of vehicles, e.g., passenger cars, motorcycles, and trucks.

Following the ideas proposed by IBM [2], one can alternatively utilize historic information, e.g., the speeds, a vehicle used on the same road before. Moreover, one can consider general information about the route like the allowed top speeds. In free flow situations, passenger cars tend to maintain these velocities [3] such that they make good predictions for average speeds. In addition, trucks can determine their average speeds based on relevant road data like gradient angles.

A third method to establish the average speed is to extend the connectivity maps by entries describing the average speeds for different vehicle types. Here, when producing a sample, a vehicle also appends its current speed to it. From that, the central server calculates the average speeds, vehicles of a certain type use in a geographical area, and adds them to the connectivity map data. A vehicle can then utilize this information to compute its own predicted average speed for the route to the border of the dead spot.

Finding the right average speed, however, is a general problem to be solved in all dead spot mitigation strategies discussed in [2, 16, 18]. Thus, when applying the data dissemination protocol as an add-on to some of these techniques, we should be able to piggyback their average speed calculation methods as well.

By dividing the distance to the end of the dead spot through the average speed, we get the time, the vehicle needs until regaining cellular network connectivity. The data dissemination protocol can now be easily amended by changing the last step shown in Fig. 3. When building up an ad-hoc network, the peers now carry out the computations discussed above and report the predicted time when they will leave the dead spot instead of the time it was entered. The vehicles send their messages to the peer that signalled the earliest time point.

The two examples depicted in Figs. 4 and 5 illustrate that the new version of the data dissemination protocol is more reliable than the old one. One reason is that the new method considers the correct route of the vehicle. For instance, in Fig. 4b, the red motorcycle sends its messages to the blue car that is much closer to the dead spot boundary than the yellow one and will therefore reach it earlier. The better average speed prediction attenuates the second problem as well. As shown in Fig. 5b, the red motorcycle detects that, in spite of the longer route to the dead spot border, it will leave it earlier than the truck since its average speed is much higher. Therefore, it keeps the messages by itself.

The use of connectivity maps is also useful to decide when ad-hoc networks should be formed. They are only sensible when a vehicle is either in a dead spot or close-by. In the latter case, it can relay messages of cars that are not connected themselves. Thus, in the new version of the protocol, ad-hoc networks are only established when a vehicle is within a certain distance to a dead spot. This is more precise than the original solution, i.e., triggering the creation of ad-hoc networks after falling below a certain signal strength [14].

In our approach, we assume that all vehicles participating in an ad-hoc network have an excerpt of the connectivity map covering their current region in place. Since we suppose that these maps do rarely change dramatically in a short amount of time, their excerpts do not need to be highly up-to-date. From our tests, we assume that downloading an excerpt every 15 or 30 minutes and, if the connectivity is bad, also more infrequently, is sufficient.

6 Conclusion

An extension to our data dissemination protocol was introduced. The amendment incorporates dead spot prediction making the decision which peer in an ad-hoc network will leave a dead spot first, more precise. This vehicle shall receive the messages to be forwarded since it will be able to submit them earlier via the cellular network than its peers. Moreover, we reported about our prototype of a dead spot prediction system and argued that such a system will be scalable. According to our estimation, the costs to produce connectivity maps for the road networks of whole countries like Norway seem to be justifiable. Nevertheless, the effort to provide the vehicles with connectivity maps is substantial such that creating them just to improve our data dissemination protocol would be unreasonable. Therefore, the amendment presented here should be combined with other dead spot mitigation strategies like those mentioned in Sect. 2.

Next, we test the prototype with various scenarios to learn more about how the data dissemination protocol can be further improved. For instance, we conduct tests to get better predictions about the volatility of the dead spots. In [15], we discuss different road trips to a remote area in the vicinity of Trondheim. We found out that the sensed round trip time at one tour was much worse than at the other ones. To establish if that was a rare event or is a regular and often observable effect, we currently measure the cellular network connectivity at fixed places. These tests shall help us to understand better how many data points have to be created to keep the connectivity maps up-to-date. In this way, the predictor can be fine-tuned to keep the data transfer, storage, and aggregation costs discussed in Sect. 3 as low as possible.

References

1. Bræk, R., Haugen, Ø.: Engineering Real Time Systems. Prentice Hall (1993)
2. DeLuca, L.S., Lakshmanan, G.T., Price, D.L., Smith, S.D.: Mitigating Service Disruptions using Mobile Prefetching based on Predicted Dead Spots (Jan 2018), U.S. Patent 9 860 336
3. Department for Transport: Free Flow Vehicle Speed Statistics: Great Britain 2012 (Jun 2013)
4. Driesen, P.E.: Prediction of Multipath Delay Profiles in Mountainous Terrain. IEEE Journal on Selected Areas in Communications **8**(3), 336–346 (2000)

5. Eltahir, A.A., Saeed, R.A., Alawi, M.A.: An Enhanced Hybrid Wireless Mesh Protocol (E-HWMP) Protocol for Multihop Vehicular Communications. In: International Conference on Computing, Electical and Electronic Engineering (ICCEEE). pp. 1–8. IEEE Computer, Khartoum, Sudan (2013)
6. Google LLC: Google Maps Services, Web Services, Direction API. https://developers.google.com/maps/documentation/directions/intro (2019), accessed: 2019-07-17
7. Jaimovich, E.: Roadways, Input Sourcing, and Patterns of Specialisation. School of Economics Discussion Papers 0118, University of Surrey, UK (2018), https://EconPapers.repec.org/RePEc:sur:surrec:0118
8. Jain, V., Raghunathan, B., Kone, V.: Dead Spot Prediction Method for Wireless Vehicular Applications (Jul 2013), U.S. Patent 8 494 563
9. Jakobsen, R.H.: Message Forwarding between Vehicles in Dead Spots. Master's thesis, Norwegian University of Science and Technology (NTNU) (May 2018)
10. Mecklenbräuker, C.F., Molisch, A.F., Karedal, J., Tufvesson, F., Paier, A., Bernado, L., Zemen, T., Klemp, O., Czink, N.: Vehicular Channel Characterization and Its Implications for Wireless System Design and Performance. Proceedings of the IEEE **99**(7), 1189–1212 (2011)
11. Meyer, J.A.E.: Dynamic Computation of Connectivity Data. Master's thesis, Norwegian University of Science and Technology (NTNU) (Feb 2019)
12. Mukherjee, S., Baid, A., Raychaudhuri, D.: Integrating Advanced Mobility Services into the Future Internet Architecture. In: 7th International Conference on Communication Systems and Networks (COMSNETS). pp. 1–8 (2015)
13. Mumtaz, S., Huq, K.M.S., Rodriguez, J.: Direct Mobile-to-Mobile Communication: Paradigm for 5G. IEEE Wireless Communications **21** (Oct 2014)
14. Puka, E., Herrmann, P.: A Data Dissemination Protocol for Vehicles with Temporary Cellular Network Inaccessibility. In: 5th IEEE International Workshop on Communication, Computing, and Networking in Cyber Physical Systems (CCNCPS). IEEE Computer, Washington, DC (Jun 2019), to appear.
15. Puka, E., Herrmann, P., Levin, T., Skjetne, C.B.: A Way to Measure and Analyze Cellular Network Connectivity on the Norwegian Road System. In: 10th International Conference on Communication Systems & Networks (COMSNETS). pp. 595–600. IEEE Computer, Bengaluru, India (Jan 2018)
16. Raghunathan, B., Jain, V.: Dead Spot Prediction Method for Wireless Vehicular Applications (Jun 2014), U.S. Patent 8 762 482
17. Siegel, J.E., Erb, D.C., Sarma, S.E.: A Survey of the Connected Vehicle Landscape — Architectures, Enabling Technologies, Applications, and Development Areas. IEEE Transactions on Intelligent Transportation Systems (2017)
18. Smith, G.J., van Leeuwen, G.W.: Mobile Communication Optimization near Wireless Dead Zone Regions (Apr 2004), U.S. Patent 6 721 572
19. Statistisk Sentralbyrå: Road Traffic Volumes. https://www.ssb.no/en/transport-og-reiseliv/statistikker/klreg (2019), accessed: 2019-07-18
20. Tseng, F.F., Filev, D.P., Makki, I.H.: Vehicular Connectivity Map (Sep 2017), U.S. Patent 9 775 128 B2
21. Tseng, F.F., Filev, D.P., Makki, I.H., Prakah-Asante, K.O., Yang, H.: Crowd Enhenced Connectivity Map for Data Transfer Intermittency Mitigation (Oct 2015), U.S. Patent 2015/028 190
22. Wi-Fi Alliance, P2P Technical Group: Wi-Fi Peer-to-Peer (P2P) Technical Specification v1.7 (2016)