# A Data Dissemination Protocol for Vehicles with Temporary Cellular Network Inaccessibility

Ergys Puka and Peter Herrmann
Norwegian University of Science and Technology (NTNU)
Trondheim, Norway
{ergys.puka, herrmann}@ntnu.no

***Abstract*—The cellular network coverage in sparsely populated and mountainous areas is often patchy. That can be a significant impediment for services based on connections between vehicles and their environment. In this paper, we present a method to reduce the waiting time occurring when a vehicle intends to send a message via a cellular network but is currently in a dead spot without sufficient coverage. We use a hybrid network approach combining cellular network access with ad-hoc networks between vehicles that are nearby. In particular, we introduce a data dissemination protocol that allows the vehicles connected through an ad-hoc network to find out which one will most likely leave the dead spot first. Messages can then be sent to this vehicle that forwards them as soon as it regains cellular network access. Further, we developed an initial implementation of this protocol using the technology WiFi Direct that is realized on many mobile phones. Implementation details of the prototype as well as analysis results regarding data transmission time limits of fast driving vehicles are discussed in the article as well.**

***Index Terms*—Cellular Network Access, Dead Spots, Data Dissemination Protocol, Ad-hoc Networking, WiFi Direct, Android.**

## I. INTRODUCTION

In the automative sector, networked applications connecting vehicles with each other and with the infrastructure have gotten popular. Such programs realize helpful services assisting drivers in various ways. For instance, by retrieving the current vehicle and road conditions, the transport safety can be improved. Furthermore, informing about the current traffic density leads to more efficient and environment-friendly driving (see [1]).

Vehicle-to-infrastructure (V2I) connections between moving vehicles and external servers are predominantly using cellular networks [2]. Thus, applications based on V2I rely on a continuous cellular network coverage which, however, is not always given in practice. This applies particularly to rural and sparsely inhabited areas and minor roads since the cell tower infrastructure is mostly oriented on the number of people sojourning in an area [3], [4]. Also mountainous terrain can lead to temporary inaccessibility since the presence of hills tends to cause echoes of signals deteriorating the radio reception [5]. On the other hand, long distance travelling through sparsely populated and often mountainous areas profits most from many of the automative applications (e.g., breakdown support, warning system about icy conditions or rocks on the roads). An extreme case is the Australian Outback where cellular network coverage can only be found around the relatively few settlements. For instance, on the 315 km long way between Uluru and Kings Canyon in the Northern Territories, that is quite popular with tourists, there is just the settlement Yulara with mobile network coverage as well as two parking lots provided with WiFi access.

A straightforward way to guarantee connectivity in the absence of cellular network coverage is satellite communication which, however, is quite expensive [6]. Therefore, we suggest a novel *data dissemination protocol* that can improve the delivery time of messages sent from vehicles in a dead spot to the external infrastructure. In this way, reports about break downs as well as important sensor data indicating the road condition can be sped up.

Our protocol capitalizes on certain spatiotemporal properties of the vehicles in an area. It utilizes mostly ephemeral ad-hoc networks between vehicles through which messages can be forwarded. The current positions of the vehicles in such an ad-hoc network are considered to find out which one will regain cellular network access first. To achieve that, the vehicles share their current cellular network signal strength and, if they are in a dead spot, the time passed since losing connectivity. If one of these vehicles is still connected, it can relay messages received via the ad-hoc network towards the mobile network. If no vehicle has cellular network access, the message is sent to the one that entered the dead spot first. Assuming that, on rural side roads and in mountainous terrain, the different vehicles tend to have similar speeds, they are for roughly the same amount of time in a dead spot. So, the one entering the dead spot first will likely leave it first as well. We will discuss this further in Sect. IV.

Shortening the waiting times until regaining cellular network access can be particularly significant in extensive dead spot areas like the Australia Outback scenario mentioned above. For instance, a vehicle that wants to send a message just after passing Yulara on its way to Kings Canyon, will have to cover a distance of about 70 km until reaching the next parking lot with network coverage. Thus, it is useful to pass the message to a vehicle moving into the opposite direction since that one will pass Yulara shortly where it can send the message. If the dead spots are smaller and a chain of vehicles goes into the same direction, one can use our approach also to forward messages along the platoon until reaching a vehicle that is outside the dead spot and can transmit them.

To show the versatility of the data dissemination protocol, we further developed a prototypical implementation using Android mobile devices connected via WiFi Direct-based ad-hoc networks [7].

The remaining article is organized at follows: After a discussion of related work in Sect. II, we give a quick overview of the WiFi Direct technology in Sect. III. Thereafter, we present the data dissemination protocol in Sect. IV and refer to realization aspects in Sect. V. In Sect. VI, we discuss some results of experimenting with our approach followed by a conclusion in Sect. VII.

## II. RELATED WORK

Given the fact that temporary cellular network inaccessibility can impede a lot of mobile applications, there exists relatively few work about dead spot mitigation. Most of the published approaches try to predict dead spots and use that information for mitigation. This proceeding seems to be highly interesting for the industrial sector as patents from IBM [8], Bosch [9], and Ford [10] illustrate. All three companies patented central server-centric approaches in which the vehicles send dead spot locations to servers that aggregate the information to what Ford calls *connectivity maps*. The aggregated data is then sent back to the vehicles that can utilize them to prepare for passing dead spot areas. As mitigation strategy, Bosch [11] and IBM [8] filed patents according to which streaming data is stored before reaching the dead spot. The additionally saved data can then be played while the vehicle has no connectivity. Knowing the extension of a dead spot, the time needed to pass it, can be predicted and the according amount of additional streaming data stored. Another IBM patent [12] describes a way to reconnect phone calls that are interrupted while a dead spot is passed. Further, the driver is informed about the reason and duration of the interruption. In this way, unnecessary attempts to reconnect can be avoided.

To get an idea about the usage of central servers aggregating connectivity data on road systems, a prototypical solution was built in a master's thesis [13]. Vehicles measure important connectivity data about cellular networks like signal strength, round trip time, and jitter in certain areas (see also [4]). In intervals, these data are transferred to a central server which aggregates the inputs from many vehicles and builds connectivity maps. The produced connectivity information is then sent back to the vehicles that can use this knowledge for mitigation. The approach showed, however, that the amount of data to be collected and aggregated will get enormous if a large number of vehicles and an extended area have to be considered. Even the relatively small number of tests conducted around the Norwegian cities Trondheim and Svolvær lead to 130,000 data samples. Scaling that up will lead to huge data centers and the transportation of vast data sets between them and the vehicles.

In consequence, looking on technology to mitigate dead spots without using centrally administrated connectivity maps seems to be worthwhile. While there exists a lot of work about using hybrids of cellular and vehicular ad-hoc networks (see, e.g., [14]), we only found one other approach using hybrid systems for dead spot mitigation. The authors of [15] do not utilize the movement patterns of the vehicles. Instead, they extend cellular network coverage by using vehicles as a relay station such that a vehicle in a dead spot can send messages via a chain of vehicles until reaching one with cellular network access. Since the transmitters of the vehicles have a limited strength and messages are not stored, this approach can only work for areas with only relatively small dead spots and sufficient traffic. For instance, for the Australian Outback example, it would not fit due to the extreme stretch of the dead spots and the low amount of traffic[1].

## III. OVERVIEW OF WIFI DIRECT

The well-known IEEE 802.11 Wireless Local Area Network (WLAN) protocol [16] is a widely used standard for home, enterprise, and public access networks. Besides other types of communication, it offers an ad-hoc mode which, however, has not been widely deployed on mobile devices. Instead, these units use WiFi Direct, a technology developed by the Wi-Fi Alliance [7] and released roughly a decade ago. An accurate and concise overview of WiFi Direct's architecture and technical details on group formation and power saving mechanisms is given in [17].

WiFi Direct is based on the IEEE 802.11 infrastructure mode which allows the user to connect WiFi Direct devices not only with each other but also with legacy WiFi devices. It represents an advance in direct device-to-device connectivity and has been implemented in most of mobile platforms. For devices supporting the Android operating system version 4.0 (API level 14) or higher, the so-called WiFi Peer-to-Peer (WiFiP2P) framework [18] is offered. It provides an implementation of WiFi Direct that complies with the Wi-Fi Alliance's certification program. Meanwhile, Apple has introduced a similar framework among iOS devices [19]. Through the adoption of these frameworks, ad-hoc or opportunistic networks of mobile devices can be relatively easily created and managed.

In contrast to other technologies (see [20]), WiFi Direct does not require the existence of a well-defined Access Point (AP) like a router. Instead, the basic network administration is taken by one of the peers. Devices seeking to connect with others, can start a discovery process. If there are other interested peers in the vicinity, a group formation process starts during which one of the stations is dynamically assigned the role of the *Group Owner* (GO). According to [7], [21], this node is the entity providing the connected clients with a Basic Service Set (BSS) functionality and related services. Further, the GO handles the integration of new nodes that will connect with an already existing network. In addition, the GO is responsible to assign other client nodes with unique IP addresses using the Dynamic Host Configuration Protocol (DHCP). The WiFiP2P framework, however, does not provide a function that lets the clients retrieve the IP addresses of the other clients. Instead,

---

[1] Sometimes, one has to wait 10 minutes or more until passing another vehicle that goes into the opposite direction.

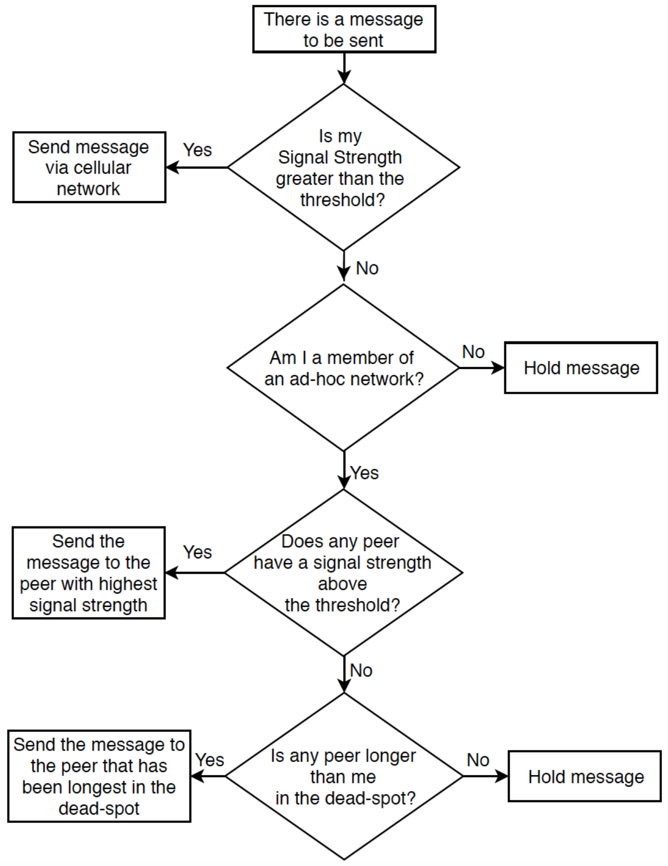


Fig. 1: Steps executed by the data dissemination protocol.

all communication shall go through the GO. As discussed in Sect. V, this detour can be avoided by forwarding the IP addresses manually. The GO also serves the network members with services reducing their power consumption.

In our work, we apply Android devices equipped with the appropriate hard- and software to run the above mentioned WiFi peer-to-peer (WiFiP2P) framework. The corresponding API makes it possible to use a set of methods to discover other devices, to request information about the discovered devices, and to connect with each other. Once the devices are connected, they can communicate by transferring data between each other through the GO. Moreover, the WiFiP2P API offers a set of Java-listeners that allows the running app to be notified of the success or failure of previous method calls. Further, Android intents are used to notify a device about specific external events like losing the connection with the GO.

## IV. DATA DISSEMINATION PROTOCOL

As discussed in the introduction, the proposed data dissemination protocol expedites the transmission and delivery of messages originating from vehicles in dead spots. To achieve that, the protocol utilizes the positions and directions of vehicles with respect to areas of cellular network connectivity. In particular, nearby vehicles build up mostly short-lived ad-hoc networks. The network peers then exchange their current signal strengths with each other. Further, if a vehicle loses cellular network coverage, it takes a time stamp. These time stamps are also exchanged with the other network peers.

If a vehicle $veh_s$ intends to send a message, it follows the steps shown in Fig. 1. First, $veh_s$ checks if it has sufficient mobile network coverage to transmit the message by itself. If that is the case, the message is sent via the cellular network. Otherwise, $veh_s$ checks if it is currently a member of an ad-hoc network. If that is not the case, the message is hold and the procedure is restarted when the station either regains cellular network coverage or joins an ad-hoc network.

If $veh_s$, indeed, is member of an ad-hoc network, it checks the signal strength values of its peers. If at least one of them has a signal strength value indicating that it has sufficient cellular network coverage, $veh_s$ sends its message to the peer with the best signal strength. That reflects the result of our tests described in [4] according to which, at least for 3G protocols like EDGE, there is a significant correlation between the signal strength and the round trip time to a remote server.

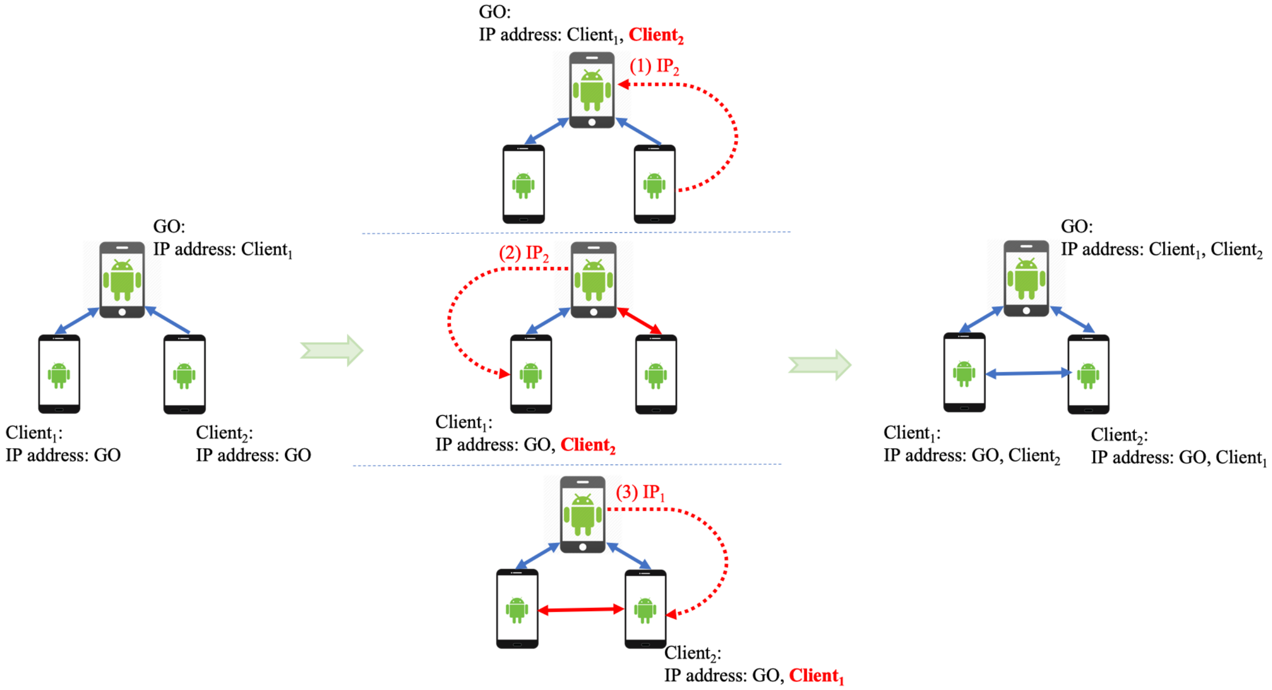
The final step utilizes the time stamps indicating when the different vehicles entered a dead spot. If, according to the time stamps, no other device in $veh_s$'s ad-hoc network has lost its connection longer than $veh_s$ itself, it holds the message assuming that it will be the peer regaining accessibility first. Otherwise, it sends the message to the peer that entered the dead spot first.

The proceeding to pass messages to peers that are longest in a dead spot holds on the following assumptions:

- *The size of a dead spot is likely the same for all vehicles passing it.* Here, the positions of the devices in the vehicles and their transmission strengths may somehow differ but that has likely only a relatively small effect on the dead spot extension.
- *The vehicles have the same average speed when passing a dead spot.* Assuming that dead spots primarily occur on minor roads and often in mountainous areas, the speeds of the vehicles are mostly determined by the road quality. Thus, the average speed will be similar for all traffic. Yet, heavy trucks which tend to be very slow at steep slopes leading to a lower average speed are an exception to this rule. In principle, one can alleviate this effect by considering the type of vehicle and its average speed but that is not followed up in the moment.

If both assumptions hold, each vehicle will need around the same time to pass a dead spot. In consequence, a vehicle that is longer in the dead spot than the others, will likely leave it earlier which is the reason for our decision to forward messages to the peer having entered the dead spot first.

Our approach does not consider road junctions in a dead spot which may lead to vastly different times until leaving it. Yet, we do not see road junctions as a major problem since there are not many roads in remote areas and, in consequence, not many junctions. Moreover, settlements were often erected at junctions. Since they are usually provided with cellular network access to support the people living there, the likelihood that a junction is in a dead spot, is reduced.



(a) $Client_2$ joins the ad-hoc network composed of $GO$ and $Client_1$.

(b) IP addresses shared among all the peers belonging to the ad-hoc network.

(c) $Client_2$ fully integrated with $GO$ and $Client_1$ in the ad-hoc network.

Fig. 2: Peer-to-peer group formation.

An approach using connectivity maps will probably result in even more precise estimations which vehicle will leave a dead spot first. The reason is that one can use the knowledge about the borders of the dead spot as well as the positions and speeds of the vehicles to compute the time, when a particular vehicle regains cellular network coverage. But, as discussed in Sect. II, this will be at the cost of big data centers and large amounts of sensor and connectivity map data to be transferred.

## V. IMPLEMENTATION USING WIFI DIRECT

We used Android mobile devices for the prototype of the protocol since they are widespread and in tendency much cheaper than more sophisticated vehicular network equipment. This decision lead to the adoption of WiFi Direct and, in particular, the WiFiP2P framework [18] to build the ad-hoc networks since this technology is supported by most Android devices. WiFiP2P offers a *WifiP2pManager* and some other supporting Java classes that proved to be very helpful for our implementation. The realization utilizes some preparatory work on WiFiP2P [22].

An issue faced during the design of an ad-hoc peer-to-peer architecture is the limitation of the framework to obtain the IP addresses of all the peers that belong to the group. Direct access between all peers is, however, important for a proper execution of the protocol. To overcome this limitation of the WiFiP2P framework, we used the solution consisting of the steps shown in Fig. 2:

1) We start with an existing network consisting of a group owner $GO$ and a $Client_1$ (see Fig. 2a).
2) When a device $Client_2$ joins this group, $GO$ provides it with a unique IP address using a built-in feature of WiFiP2P. As shown on the top of Fig. 2b, $Client_2$ sends its IP address to $GO$ that stores this information in a *Java HashMap*[2].
3) In the next step depicted in the center of Fig. 2b, $GO$ sends the address of $Client_2$ to all other peers (i.e., $Client_1$ in our example) that store it in their own hash maps. Thereafter, the new device $Client_2$ is known to the other devices and can be reached from them.
4) As shown at the bottom of Fig. 2b, $GO$ sends the IP addresses of all the other peers to $Client_2$ that now can reach all other members of the ad-hoc network directly.
5) $Client_2$ is now fully integrated with $GO$ and $Client_1$ into the ad-hoc network which is depicted in Fig. 2c.

Likewise, peers leaving a network since they come out of range, have to be considered. WiFi Direct offers a method informing the GO when a client peer is lost. The GO then removes the hash map entry of that peer and informs the remaining clients which can adapt their hash maps accordingly. If there are no other clients left, the ad-hoc network is dissolved and the station realizing the GO can start discovery to find other connection possibilities. When a client loses the connection to the GO, it receives a network termination mes-

---

[2] We have to use this data transfer since, albeit $GO$ creates the IP addresses of the clients, WiFiP2P does not offer a method to retrieve them directly.

sage [7]. It can then empty its hash map and start discovering to build a new ad-hoc network.

The mechanisms described above guarantee that all peers in the ad-hoc network hold the required information to send messages directly to any other peer in the group. While the GO is the entity responsible for the group stability and conducts amongst others the IP address management, all devices in the network are equally important peers with respect to the data dissemination protocol. Therefore, we provide all peers with a full set of IP addresses of all the other ones which allows them to forward messages directly to their destination instead of having to relay them via the GO. We discuss in Sect. VI that the IP sharing phase is usually 0.5 seconds while the average time for the message delivery is about a second since we use message confirmation. These short times can prove very helpful for fast driving opposite traffic. For instance, the maximum connection time is 6.5 seconds if both vehicles have a speed of 110 km/h.

Another important feature of our data dissemination protocol discussed in Sect. IV is to inform all peers about the quality of the mobile network coverage. In Android, Arbitrary Strength Unit (ASU) and dBm (power ratio in decibels per milliwatt) are the two ways to indicate the quality of a cellular network. We use ASU which allows us to characterize the signal strength by integer values between 0 and 31 resp. 99 when the signal strength cannot be determined, e.g., for technical reasons [4]. The ASU value is obtained by using a set of built-in listeners of the Android *ConnectivityManager* class that notify the application whenever the ASU value changes. When a device is being notified about a change of the signal strength, it informs all peers in its ad-hoc network. The devices use an entry for the signal strengths of their peers in their hash maps which is updated after receiving this change information.

Moreover, if a device loses its cellular network connection completely, it saves the moment when that occurs as a standard *Java Timestamp*. To avoid clock synchronization problems, the peer sends the duration since entering the dead spot to the other peers in the ad-hoc network. These subtract the received duration from their own time stamp taken when receiving the message and store the result in their hash maps. Disregarding communication delays, the time stamps of the different peers can then be compared without having to deal with clock synchronization issues.

With all this information in place, a device which wants to transmit a message via the cellular network, follows the data dissemination protocol. If it has sufficient mobile network access, it sends the message. Otherwise, it retrieves from its hash map whether peers in the ad-hoc network are connected to the mobile network. If that is the case, the message is forwarded to the station with the highest ASU value that can immediately forward the data package via the cellular network. If no peer has mobile network coverage, the one with the oldest time stamp is retrieved. If that peer is longer in the dead spot than the message holder itself, the message is sent to it assuming that the peer will also be the first one regaining mobile network access. As soon as the recipient of the message leaves the dead spot, it can then forward the package via the cellular network. In this way, the fastest possible message delivery is achieved.

To reduce the building of unnecessary ad-hoc networks and to save energy, we further stop the discovery when the ASU of a station is above a certain threshold showing real good cellular network coverage. In that case, it is assumed that the other stations in the vicinity will also have significant mobile network accessibility such that the data dissemination protocol is not yet needed.

## VI. EXPERIMENTAL RESULTS AND STATISTICS

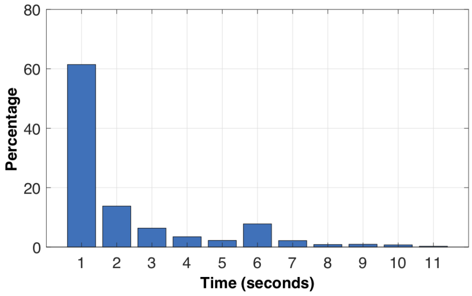
To evaluate our data dissemination protocol in dead spots, we have set up a testbed composed by two Moto $Z^2$ Play smartphones from Motorola and one Galaxy Tab A tablet from Samsung. Both smartphones run on Android Nougat OS, version 7.1.1, while the tablet uses Android Oreo OS, version 8.1. We conducted several experiments with different distances and movements to get an idea about the performance of WiFiP2P [18], in particular, the times needed to discover the ability of building an ad-hoc network, to connect oneself into a network, and to carry out the data dissemination protocol described in Sect. IV.

The most time-critical case is exchanging messages with traffic going into the opposite direction. According to [23], the maximum transmission distance in WiFi Direct technology is 200 meters[3]. Moreover, this distance was confirmed for moving vehicles using the underlying technology IEEE 802.11 [24], [25]. Since a vehicle $veh_s$ can start building an ad-hoc network when the opposing traffic $veh_o$ is 200 meters ahead and can communicate until it is 200 meters behind $veh_o$, we can calculate the available time as 400 meters divided by the added speeds of $veh_s$ and $veh_o$. Thus, on bad roads on which both vehicles have an average speed of only 50 km/h, they have 14.4 seconds to build up the ad-hoc network and to send messages. In Norway, the general speed limit outside of inhabited areas is 80 km/h. If both, $veh_s$ and $veh_o$ have this speed, the available time for handling the data dissemination protocol is nine seconds. If the two vehicles have speeds of 110 km/h that are allowed on some Norwegian motorways and also on many roads in the Australian Outback, the disposable time is only 6.5 seconds.

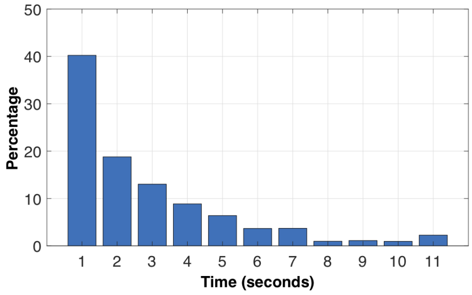
As mentioned above, the message exchange consists of the discovery, connection, and data dissemination phases. Altogether, we conducted 39,896 measures to find out the duration of executing the discovery phase. Further, we carried out 3,585 measurements of the connection phase length and measured the required time to send a message of up to 256 bytes length among peers in 2,393 cases. In Fig. 3, we list the results of our experiments in the form of bar diagrams. Here, each bar depicts the percentage of the overall number of experiments that were conducted in a certain time interval measured in seconds. To keep the diagrams short, we list all cases in

---

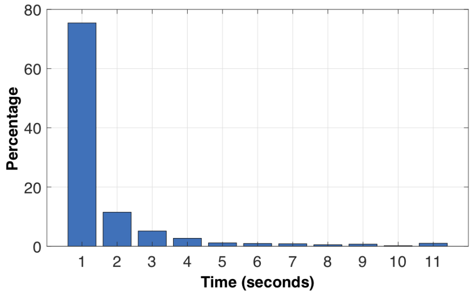
[3] Our own tests revealed that even 250 meters distance allows ad-hoc network building and message transfer without problems.



(a) Discovery phase.



(b) Connection phase.



(c) Message delivery phase.

Fig. 3: Average time durations of the three protocol phases.

which a phase could not be finished within 10 seconds, in the 11 seconds bar. The results are as follows:

- The diagram in Fig. 3a depicts that devices discovers other ones in their vicinity in roughly 61% of the cases within one second and 74% within two. The average of the delay is 1,810 ms and the median 772 ms.
- The diagram in Fig. 3b displays the times, two nodes, which have already discovered each other, need to create a new ad-hoc group or join an existing WiFi Direct group to which one of the peers already belong. This phase is slower than the discovery since group management negotiations, the security protocol WiFi Protected Setup (WPS), and the DHCP have to take place [26]. The average delay is 2,287 ms and the median 1,409 ms. In roughly 71% of all cases, nodes can be connected within three seconds.
- Since the data transfer between the nodes to hand over messages for faster transmission via the cellular network is confirmed, the diagram in Fig. 3c presents the times to send both, a message and a confirmation packet indicating that the message has successfully reached the destination node. On average, this process takes 967 ms and the median 306 ms. Moreover, we can observe that in 76% of all cases, the data transfer is finished within one second and in 86% within two.
- Another time factor to be added, is the delay needed to share the IP addresses which is described in Sect. IV. We did not produce an own diagram for that since it resembles the one in Fig. 3c albeit with half the time as the IP exchange messages are not confirmed.

The significant difference between the average and median values in all three phases points to the fact that there were some few very bad results that deteriorated the averages.

If we consider the three phases as independent, the likelihood to achieve all of them within the 6.5 seconds needed when both vehicles have a speed of 110 km/h, is 71%. When the cars have a speed of 80 km/h which gives us a time interval of nine seconds, the message exchange is successful in around 86% of all cases while in the case of a speed of 50 km/h and a duration of 14.4 seconds, the success rate adds up to 97%.

Yet, assuming the three phases as independent seems to be overly conservative since long execution times detected in one phase are probably caused by a connectivity issue that will also take effect in the other phases. In contrast, if the connection phase as the most time-critical one works fast, one can assume that the other two phases can be also carried out smoothly such that the likelihoods of successful message exchanges should be higher. To find out if that is indeed the case, we also tested the three phases in dependency by counting the time necessary to carry out the whole process from starting the discovery until completing the confirmation for transmitting a message. In this case, 71% of all tests were finished in 6.5 seconds, 97% in nine seconds, and all tests in 14.4 seconds. Thus, there seems to be in fact dependencies between the three phases that, however, only take effect for time intervals greater than seven seconds.

From our tests, we can now argue that the realization of the data dissemination protocol using WiFiP2P is very reliable for vehicle speeds until 80 km/h. If the vehicles are faster, however, the likelihood of failures due to slow ad-hoc network building rises. Nevertheless, even then the implementation seems to be useful since when a vehicle with a speed of 110 km/h misses to hand a message over to one passing it in opposite direction, it will try to transfer it to the next vehicle. Assuming a likelihood of 71% for successful transmissions, the probability that a message can be handed off to one of three vehicles going into the other direction, is around 98% which is quite solid.

One of the first works about using the WiFi Direct protocol to create ad-hoc connections between different Android devices close to each other, is presented in [26]. In their article, the authors analyzed and evaluated the times required for the different phases of the WiFi Direct group formation, but without considering the stages of the peer-to-peer group formation discussed in Sect. V. Comparing our results with theirs show that the discovery and group formation phases need less time in our case. One reason is probably that today's equipment is more powerful than those available in 2013. Another one is the fact that their scenario implies the simultaneous group formation of at least three devices. In contrast, our approach is mostly incremental. Usually, an ad-hoc network is built of two vehicles meeting each other while further ones are only added if they come closer to the existing group members. Thus, the negotiation effort is usually smaller.

## VII. Concluding Remarks

In his paper, we propose a data dissemination protocol combining ad-hoc and cellular networks to accelerate the V2I transmission of messages originating from vehicles that are in a dead spot. Moreover, we developed a prototype of this protocol adopting the widespread technology WiFi Direct [7].

Our tests revealed that, except for very high speeds, WiFiP2P, the WiFi Direct realization of Android, is fast enough to guarantee the timely transfer of messages between vehicles passing each other in opposite direction. Nevertheless, to cover also speeds of 110 km/h or faster with a good reliability, it seems worthwhile to look at alternative technologies as well. For instance, WiFi Aware [27] is a new technology provided by the Wi-Fi Alliance. It promises faster device discovery and network building. Since these two phases are the most time consuming parts in our approach, WiFi Aware might be a better fit than WiFi Direct. Likewise, we will look on vehicular ad-hoc networks (VANET, see [2]) which, in the close future, will be used in many vehicles. With their range of up to a few kilometers, they may also be a good platform for the data dissemination protocol.

Further, we are also thinking of enabling the transmission of messages starting in the infrastructure and going towards vehicles with temporary cellular network inaccessibility. Here, a novel forwarding mode called GeoCast (see [1]) might be helpful. It is a position-based routing mechanism that sends messages to all devices in a certain geographical region. Messages towards a vehicle $veh_r$ that might not be reachable but which approximate position is known, can be sent into its area using GeoCast. Then other vehicles nearby with mobile network access can store the message. As soon as they are linked with $veh_r$ in an ad-hoc network, they deliver the message to this vehicle. In areas with large dead spot areas like the Australian Outback scenario such an extension may speed up the reception of messages by the vehicles significantly.

## References

[1] P. Papadimitratos, A. de La Fortelle, K. Evenssen, R. Brignolo, and S. Cosenza, "Vehicular Communication Systems: Enabling Technologies, Applications, and Future Outlook on Intelligent Transportation," *IEEE Communications Magazine*, vol. 47, no. 11, pp. 84–95, 2009.

[2] 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.

[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] E. Puka, P. Herrmann, T. Levin, and C. B. Skjetne, "A Way to Measure and Analyze Cellular Network Connectivity on the Norwegian Road System," in *10th International Conference on Communication Systems & Networks (COMSNETS)*. Bengaluru, India: IEEE Computer, Jan. 2018, pp. 595–600.

[5] 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.

[6] A. G. C. Guerra, A. S. Ferreira, M. Costa, D. Nodar-López, and F. A. Agelet, "Integrating Small Satellite Communication in an Autonomous Vehicle Network: A Case for Oceanography," *Acta Astronautica*, vol. 145, pp. 229–237, 2018.

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

[8] L. S. DeLuca, G. T. Lakshmanan, D. L. Price, and S. D. Smith, "Mitigating Service Disruptions using Mobile Prefetching based on Predicted Dead Spots," U.S. Patent 9 860 336, Jan. 2, 2018.

[9] V. Jain, B. Raghunathan, and V. Kone, "Dead Spot Prediction Method for Wireless Vehicular Applications," U.S. Patent 8 494 563, Jul. 23, 2013.

[10] F. Tseng, D. P. Filev, I. H. Makki, K. O. Prakah-Asante, and H. hsiang Yang, "Crowd Enhenced Connectivity Map for Data Transfer Intermittency Mitigation," U.S. Patent 2015/028 190, Oct. 1, 2015.

[11] B. Raghunathan and V. Jain, "Dead Spot Prediction Method for Wireless Vehicular Applications," U.S. Patent 8 762 482, Jun. 24, 2014.

[12] G. J. Smith and G. W. van Leeuwen, "Mobile Communication Optimization near Wireless Dead Zone Regions," U.S. Patent 6 721 572, Apr. 13, 2004.

[13] J. A. E. Meyer, "Dynamic Computation of Connectivity Data," Master's thesis, Norwegian University of Science and Technology (NTNU), Feb. 2019.

[14] S. Mukherjee, A. Baid, and D. Raychaudhuri, "Integrating Advanced Mobility Services into the Future Internet Architecture," in *7th International Conference on Communication Systems and Networks (COMSNETS)*, 2015, pp. 1–8.

[15] A. A. Eltahir, R. A. Saeed, and M. A. Alawi, "An Enhanced Hybrid Wireless Mesh Protocol (E-HWMP) Protocol for Multihop Vehicular Communications," in *International Conference on Computing, Electical and Electronic Engineering (ICCEEE)*. Khartoum, Sudan: IEEE Computer, 2013, pp. 1–8.

[16] IEEE Standards Organization. IEEE 802.11-2016. [Online]. Available: https://standards.ieee.org/standard/802_11-2016.html

[17] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano, "Device-to-device Communications with Wi-Fi Direct: Overview and Experimentation," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 96–104, 2013.

[18] Android Developers. Wi-Fi Peer-to-Peer. [Online]. Available: https://developer.android.com/guide/topics/connectivity/wifip2p

[19] iOS Developer Library. Multipeer Connectivity Framework. [Online]. Available: developer.apple.com/documentation/multipeerconnectivity

[20] J. Lee, Y. Su, and C. Shen, "A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," in *33rd Annual Conference of the IEEE Industrial Electronics Society (IECON)*, Nov. 2007, pp. 46–51.

[21] C. Funai, C. Tapparello, and W. Heinzelman, "Enabling multi-hop ad hoc networks through WiFi Direct multi-group networking," in *International Conference on Computing, Networking and Communications (ICNC)*, Jan. 2017, pp. 491–497.

[22] R. H. Jakobsen, "Message Forwarding between Vehicles in Dead Spots," Master's thesis, Norwegian University of Science and Technology (NTNU), May 2018.

[23] D. Feng, L. Lu, Y. Yuan-Wu, G. Y. Li, S. Li, and G. Feng, "Device-to-device communications in cellular networks," *IEEE Communications Magazine*, vol. 52, no. 4, pp. 49–55, April 2014.

[24] J. Ott and D. Kutscher, "Drive-thru Internet: IEEE 802.11b for "Automobile" Users," in *IEEE INFOCOM*, Hong Kong, 2004.

[25] R. Gass, J. Scott, and C. Diot, "Measurements of In-Motion 802.11 Networking," in *7th IEEE Workshop on Mobile Computing Systems & Applications (WMCSA)*. IEEE Computer, 2006, pp. 69–74.

[26] M. Conti, F. Delmastro, G. Minutiello, and R. Paris, "Experimenting Opportunistic Networks with WiFi Direct," in *IFIP Wireless Days (WD)*, Nov. 2013, pp. 1–6.

[27] Wi-Fi Alliance, P2P Technical Group, "Wi-Fi Certified Wi-Fi Aware$^{TM}$ Technological Overview," 2018.