A vehicle-to-vehicle communication protocol for collaborative identification of urban traffic conditions

Øyvind Risan[1], Evtim Peytchev[2]

[1]Norwegian University of Science and Technology, Åbyfaret 141, N-1392 Vettre, Norway, risan@item.ntnu.no
[2]Nottingham Trent University, Computing and Informatics Building, Clifton Lane, Nottingham, NG11 8NS, United Kingdom, evtim.peytchev@ntu.ac.uk

Abstract: This paper proposes a vehicle-to-vehicle (V2V) communication protocol which makes it possible to discover and share traffic status information in a novel, efficient and comprehensive way. The protocol is specifically designed to work in an environment without infrastructure where all the vehicles (nodes) can talk to each other (ad-hoc network) and collaboratively generate new knowledge relevant to the traffic conditions existing at that moment in an urban environment. The nature of such a network demands self-configuration and autonomous behaviour. The protocol adheres to these principles and makes it possible for the nodes to initiate discovery and determine the location of areas where specific traffic conditions apply. The proposed "Single Ripple" algorithm determines these areas by only involving vehicles with the desired conditions and their neighbours. The algorithm imposes only a minimal load onto the wireless network.

Keywords: Traffic information, ad-hoc networks, area discovery, ubiquitous networking, traffic conditions communication protocol, vehicle-to-vehicle communication, V2V, collaborative (cooperative) wireless traffic information systems.

1. Introduction

1.1. Background

Global transportation problems are becoming more difficult to solve year after year as the complexity of the traffic network and the number of vehicles on the roads increase. Research in the area of demand-responsive traffic control already deals with many complex problems [1] and vehicles in their own right are becoming more and more complex – for example, in a modern car, several hundred sensors are used to keep the car working properly. It is also widely accepted that the majority of the modern vehicles are well equipped with computation and communication hardware. The big question now is how to harness the potential benefits of this new in-car resource to the full benefit of the driver, the transportation system and society [2], [7], [8]. So far, system-wide solutions to urban transportation problems have relied on centralised traffic control. These solutions have worked well in the context of previous generation telemetry systems but are bound to be overtaken by a new generation of solutions enabled through a new generation of protocols based on the principles of ad-hoc wireless communication networking. Existing algorithms that provide routing information throughout the whole network are not applicable, as these involve every node within the network. Similar algorithms that are based on point-to-point would fail to discover the boundaries of a condition efficiently with the desired flexibility.
After equipping a vehicle with communications technology it will no longer be an isolated node but become a part of a bigger system. Exchange of information between vehicles might prevent accidents and increase safety & efficiency [3], [6], [9], [10]. The information exchange, however, is on a peer-to-peer basis as opposed to sending all the information through a central point (server). By gathering the available information in such manner it is possible to provide new services and discover new functionalities in the wireless systems.
Some countries have introduced a so called "Zero-Road-Fatality-Vision" policy [1]. The aim of these policies is to reduce the number of traffic related-fatalities to a minimum. The implementation of technology enabling vehicle-to-vehicle communication can provide a valuable contribution to achieving this vision.

1.2. Problem description

The research questions we are trying to answer in this paper are:

- Given the vast amount of vehicles in our cities, can we utilise their communication and computing power to collect and distribute traffic information in an efficient manner?
- What traffic conditions can we discover using vehicle-to-vehicle communication and collaboration?
- Can we use information available in a single vehicle, combined with the information in other single vehicles, to detect traffic conditions?
- How big is the area where these traffic conditions exist?

Such traffic conditions might be ice on the road, fog, rain, snow, grid lock and so on.

2. The ad-hoc (peer-to-peer) networking approach

2.1. Basic description of the system

Vehicle-to-vehicle communication can be considered a form of communication in a mobile ad-hoc network, often with multiple hops. The fact that the sender and the receiver are placed in a vehicle, that can reach speeds far beyond any pedestrian, presents some challenges. At low speeds there are few location changes per second, but as speed increases the number of location changes per second rises dramatically. This fact demands that the system must be robust and self configuring.

A system might consist of just two vehicles, and cover a very small geographical area. The system might also consist of hundreds of vehicles and cover a huge area. The density of vehicles and the location of each vehicle will determine the size of the covered area. Due to limitations on the transmitted power, and signal propagation, there will be a limit as to how far two adjacent vehicles can be apart before the system must be considered to be two autonomous systems, or sub-systems.

The system will only exist as long as there are vehicles that have information to exchange, and as long as there are vehicles available to keep the information alive. If there are no vehicles in the system, there is no information of interest to be exchanged, and the need for the system disappears. Figure 1 describes how vehicles (both cars and buses) might communicate with each other, and with infrastructure.

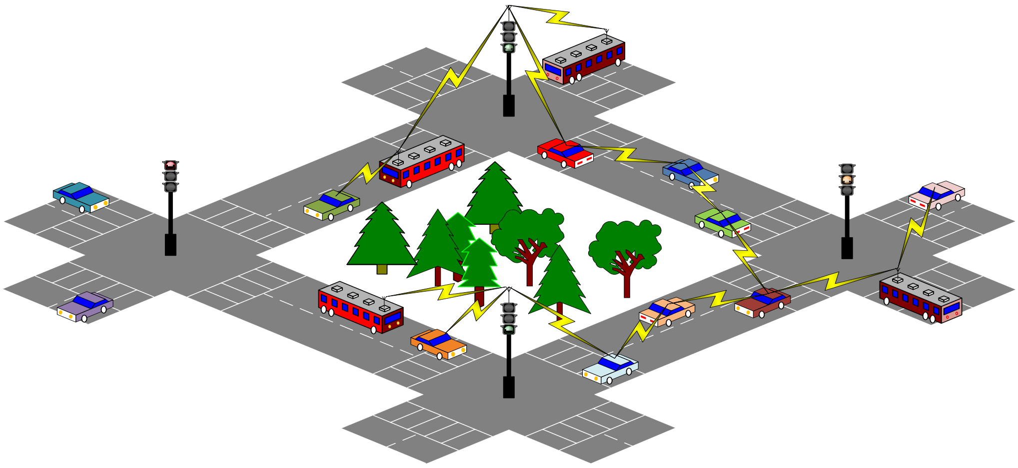


Figure 1 Vehicles and infrastructure working together

2.2. Message exchange assumptions

A transmission of messages relevant for the "collaborative traffic condition area discovery" can be conducted in two ways: through retransmission either to specific addresses or to everyone. The latter method of communication will in this paper be referred to as "pure flooding".

Transmission to specific addresses is most effective if the environment and the receivers are static, or motionless. Once the discovering and determination of neighbours are finished, each message needs only be sent once, which saves capacity and time. This solution might demand special hardware such as directive antennas and controllers.

3. Protocol description

3.1. Protocol description - area discovery

In order to discover how many vehicles have a specific condition, or the size of the condition the protocol utilises a special algorithm called "Single Ripple". The essence of this algorithm is to have a single vehicle act as a trigger for the area discovery process, which normally would be the first vehicle to identify the existence of the traffic condition (e.g. slippery surface). It issues a request for area discovery which is retransmitted as a "lake wave" or a "ripple" geographically directed outwards across all vehicles. As soon as the message reaches a vehicle without the specified condition, the message (the "ripple") is sent back (bounced) to the originator of the message. Analysing the GPS positioning of all vehicles that bounced the message provides the boundaries of the traffic condition area that is to be discovered.

The minimal format of the discovery message can be constructed as illustrated in Figure 2.

The field "Direction" indicates in which direction[1] the message is being transmitted. It is sufficient to identify whether the message is "outwards" (away from the originator), or if it is a "return" (heading back to the originator as a reply). This field should only be changed by an originator, or a replier (bouncer).

The content of the "Reply-sender + Status" field is dependent on the direction in which the message is travelling. In an outwards message the originator uses its own address and conditions, while in a return message the replier places its address and status in this field. This way, the intermediate vehicles can update their information tables with as much information as possible.

Timestamp (integer)	Originating sender + Condition (integer + hex)	Position (decimal)	Direction (binary)	Reply-sender + Status (integer + hex)

Figure 2 Structure of a discovery-message

There are two ways to determine the area of a condition. One can find all the vehicles that have the condition, and determine the area by comparing their positions. This will give a very accurate description about where the conditions apply, except for the fact that it fails to discover the borders. The result is an underestimate of the size, and would be on the form: "The area is at least this big". This information is interesting, but not as useful as a result on the form "The area is smaller than this".

By constructing the discovering algorithm carefully we can reduce the amount of uncertainty regarding the result. The information we want is the position of vehicles that do not have the condition, but which have a neighbour that does. The border of the condition then has to be between these two vehicles.

A rain cloud can be used as an example, since it will illustrate the area determination quite clearly. The function of area determination is of course applicable to many other fields and conditions. This paper only focuses on getting information from the vehicles; it does not explore how this information could be utilized.

Figure 3 illustrates 5 vehicles[2] inside a rain cloud. These vehicles communicate with each other, and with the vehicles[3] just outside the cloud. The vehicles on the outside of the cloud respond by returning their positions, and information that they do not have rain, to the vehicles inside the cloud. The vehicles inside the cloud update their information tables, and pass this information to the other vehicles inside the cloud. Based on the data in the information tables the vehicles can calculate the area that the cloud covers.

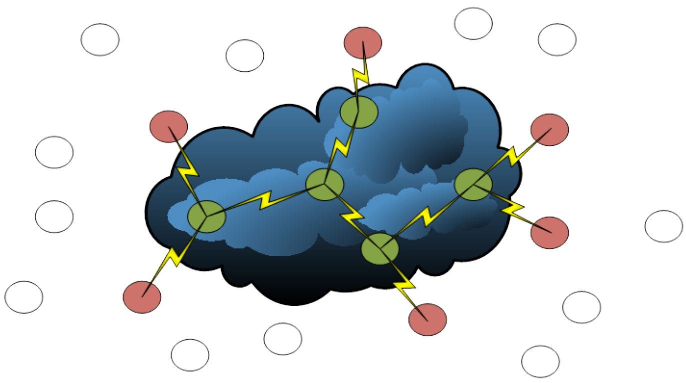


Figure 3 Vehicles determining the size of a raincloud

[1] Not in a geographical sense, but outwards from, or returning to the originator
[2] Green circles
[3] Red circles

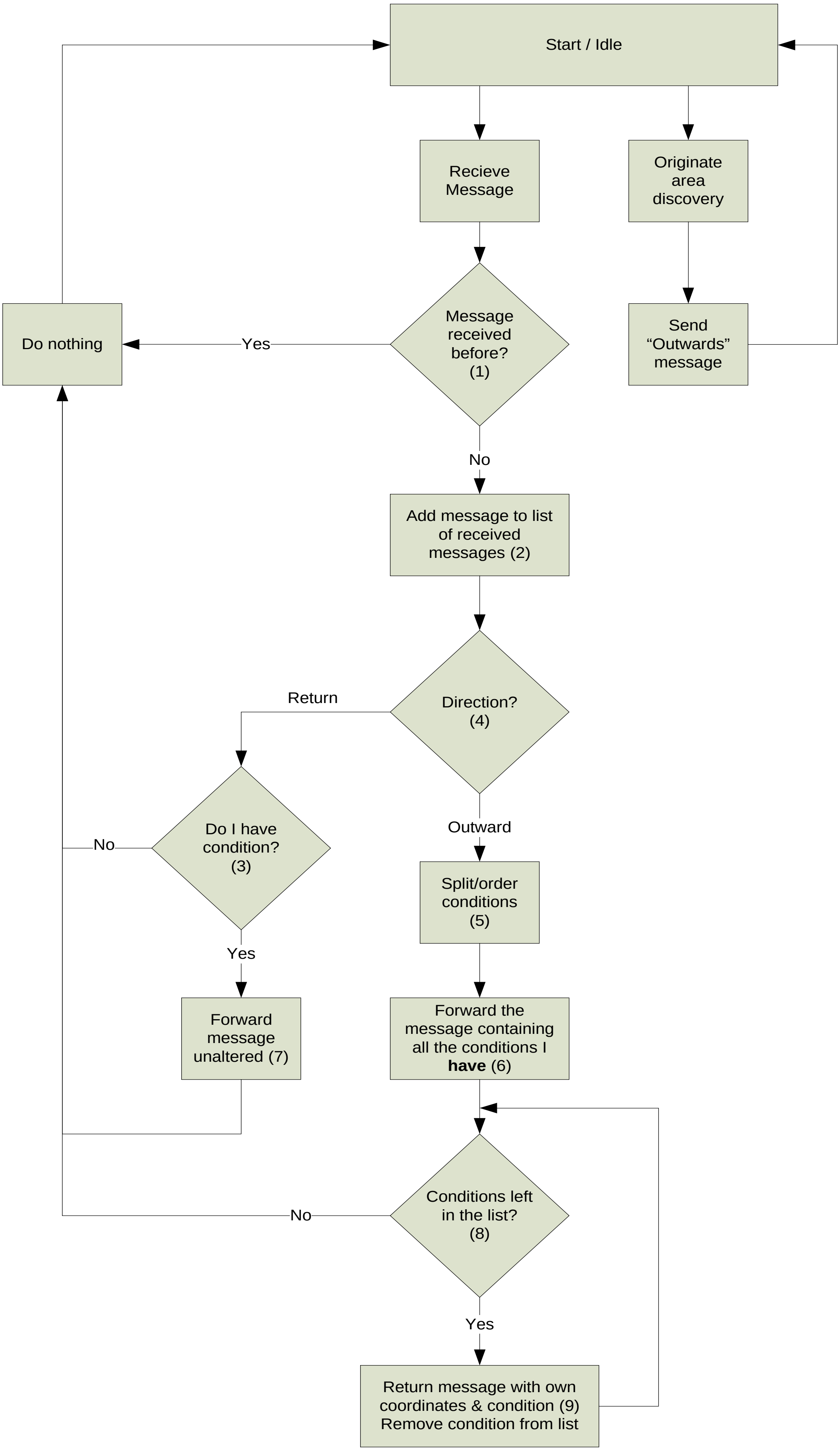


Figure 4 Algorithm for discovering areas

4. Implementation

4.1. Area discovery

Figure 4 illustrates the “Single Ripple” algorithm for message handling when a condition area is discovered. Each vehicle starts off in the “start/idle” state before it migrates through the various states and actions, until it ends up back in “start/idle” again.

The algorithm presented in Figure 4 illustrates the originator deciding to initiate an area discovery, and how every other vehicle reacts when it receives such a message. This algorithm allows the originator to inquire about multiple conditions at the same time. This will make it possible to discover the size of several areas with one message.

A scenario that can be used as an example is the one presented in Figure 5. This scenario consists of four vehicles and two conditions (rain and ice).

In this scenario the connections are as follow: 0->(1), 1->(0,2), 2->(1,3), 3->(2).

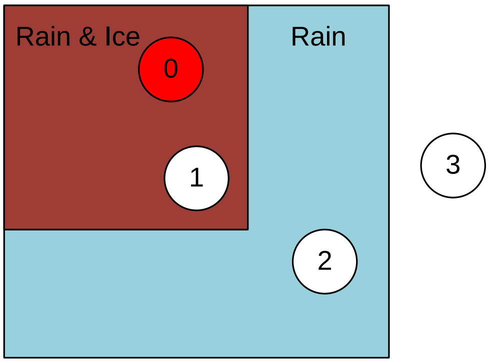


Figure 5 Example of two conditions scenario

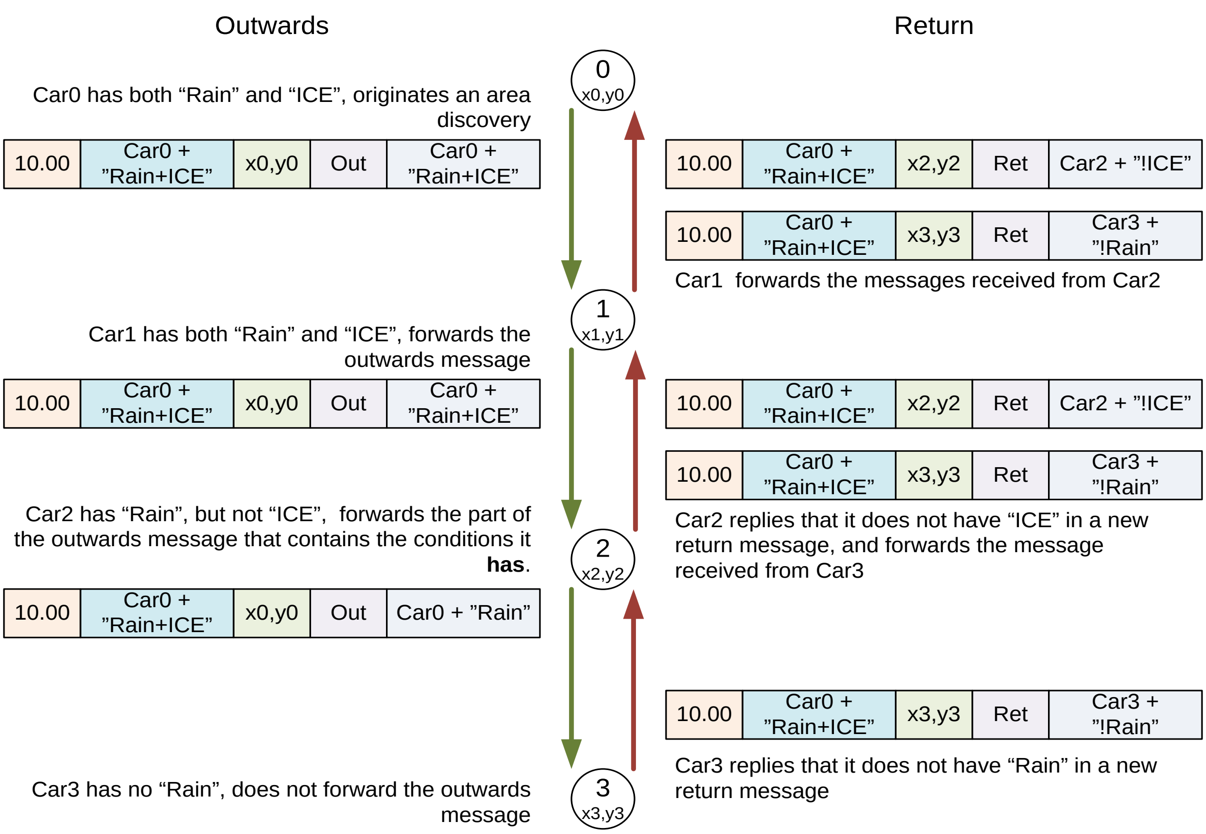


Figure 6 Example of a Discovery-message flow

The area discovery is initiated by Vehicle 0. The aim is to obtain information about the respective areas. In this example, the number of vehicles (0 to 3) is kept at a bare minimum to keep the amount of messages manageable. The vehicles are positioned so as to illustrate as many different combinations of behaviour as possible. The flow of messages is presented in Figure 6. In this example, no duplicate messages are shown, and neither are messages that are redundant.
The red vehicle (Vehicle 0) starts off the discovery by sending out a query about Rain- and Ice-area. A vehicle should discard any message that has either been seen before or that is without relevance to that vehicle. There are two reasons for a message to be of no relevance to the vehicle: The vehicle has already seen another version of the same message, or the vehicle is "outside" the area of interest, and will correctly discard any "return" messages.
In accordance with the algorithm given in Figure 4 the "Timestamp" and "Originator+Condition." remain unaltered at all times. These fields identify messages related to the query originated by "Vehicle 0". The notation (!Ice) means ("not Ice"), and represents the fact that the vehicle is not within the Ice area.

5. Simulation

The hypothesis is:

> *The "Single Ripple" algorithm will perform faster, and with a smaller load on the system than would "pure flooding".*

The goal is to test how the algorithm in Figure 4 would perform against a standard "pure flooding" algorithm. By recording the performance of each algorithm it is possible to compare how different numbers of vehicles affect the algorithms.
Relevant measurements:

- How many packages were sent.
- How many packages were received.
- Time between first and last package received.
- How many packets where dropped due to interference.

The scenario used in the simulation can be seen in Figure 7. It is similar to the examples presented earlier in this paper, but it contains 16 vehicles and two semi-overlapping conditions. The vehicles are given conditions based on their location. Vehicle 0 and 1 have two conditions, vehicle 2 and 5 have one condition (but not the same); all other vehicles do not have any conditions.
The placement, and numbering, of the vehicles allow testing with 4, 8, 12 and 16 vehicles without having to change their positions.

To test with 4 vehicles simply remove vehicles 4 – 15.
To test with 8 vehicles simply remove vehicles 8 – 15.
To test with 12 vehicles simply remove vehicles 12 – 15.
To test with 16 there is no need to remove any vehicles.

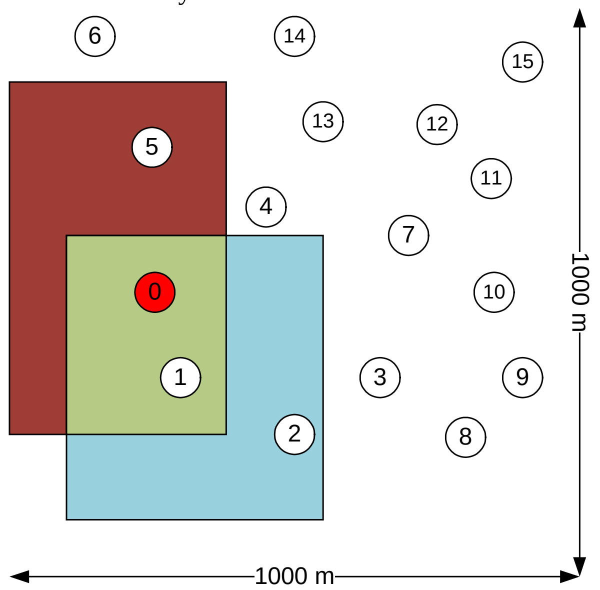


Figure 7 Scenario for the simulator

The distances between the vehicles are rather large[4]; this is done to make the simulator work under as difficult conditions as possible. The simulation is based on standard 802.11 communications, as this is the most widespread technology, and a very likely candidate for the communication between the vehicles.
The simulation was performed with immovable vehicles. The main concern about simulating with static vehicles is that this ignores the difference in interference patterns you would expect with any change in inter-vehicular positions. However one can assume that the distance each vehicle moves during the time it is actually involved in the communication is so short that its movement will not significantly influence the pattern of interference.
From the simulation result one can see that the longest average time any one vehicle was involved in the communication is 44.1 ms, at “pure flooding” and “16 vehicles” (Table 1). The average speed in some major cities in the UK is just 17.8 mph (= 28.2 km/h). An average speed of 30 km/h means that a vehicle will travel 8.3 meters each second. In 44.1 ms the vehicle would then have moved 36.6 cm. This movement is so small that it can be neglected in this simulation. At a speed of 90 km/h the vehicles will move 1.1 meter in 44.1 ms.

6. Results

The results gathered from the simulator are displayed in Table 1. The table has four main parts, one for each group of vehicles (16, 12, 8 and 4). Each of these groups is presented with results for two scenarios: “*Single Ripple*” algorithm and “pure flooding”.

	16 Vehicles		12 Vehicles		8 Vehicles		4 Vehicles	
	"Ripple"	Pure	"Ripple"	Pure	"Ripple"	Pure	"Ripple"	Pure
Avg time	4.00	44.10	5.50	31.20	6.90	17.30	4.00	7.80
TX	16	143	18	89	20	44	10	16
RX	26	282	37	157	41	80	16	24
Dropped	16	143	9	74	7	16	0	0
Total	42	425	46	231	48	96	16	24
D/Total	38.10 %	33.65 %	19.57 %	32.03 %	14.58 %	16.67 %	0.00 %	0.00 %
D/Vehicle	1.0000	8.9375	0.7500	6.1667	0.8750	2.0000	0.0000	0.0000
D/TX	4.0000	3.2426	1.6364	2.3718	1.0145	0.9249	0.0000	0.0000
D/RX	0.6154	0.5071	0.2432	0.4713	0.1707	0.2000	0.0000	0.0000

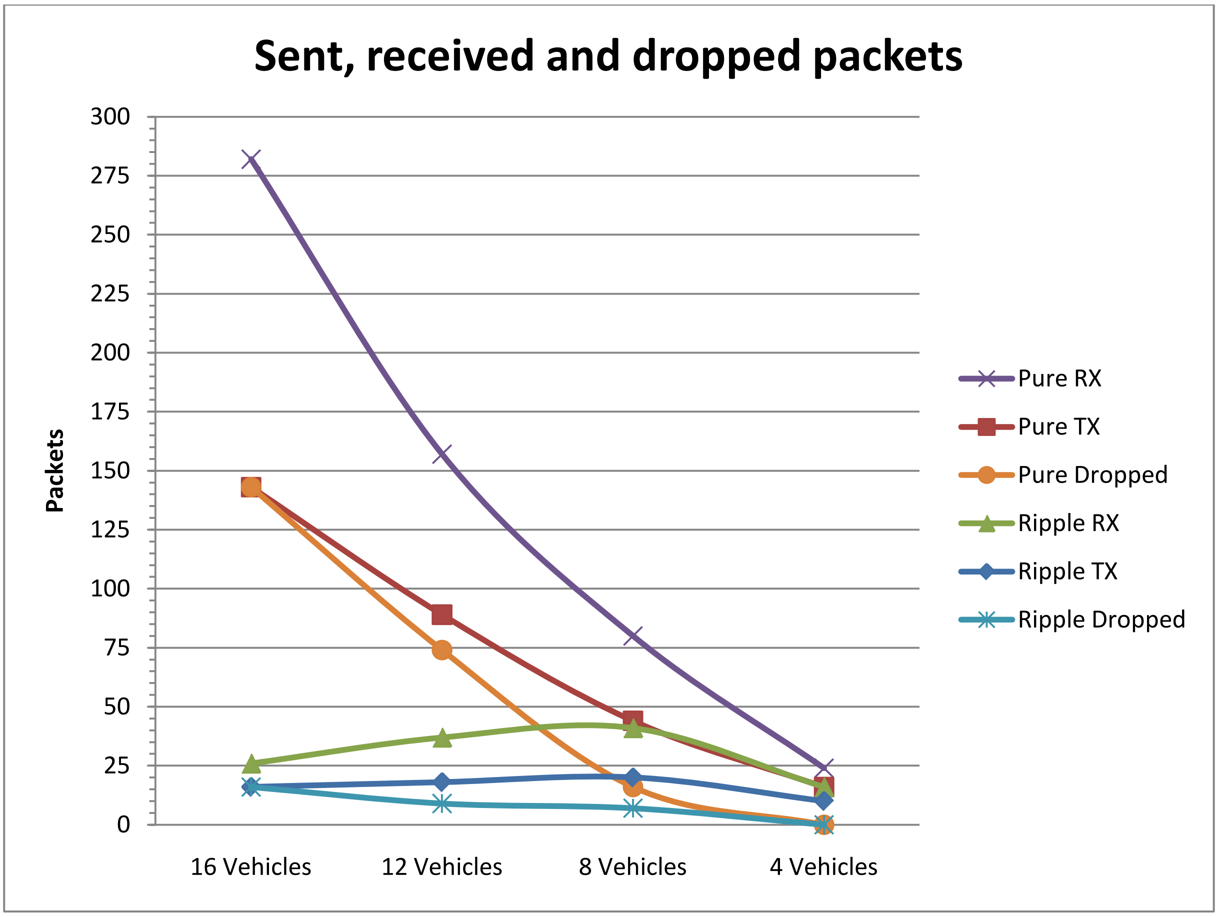
Table 1 Results from the simulator with 16, 12, 8 and 4 vehicles

- “Avg time” [5] refers to the time between the first and last message at each vehicle. The average time is based on vehicles that are actually involved, leaving out any vehicles that have received 1 packet or less.
- “TX” refers to the number of messages sent by each vehicle.
- “RX” refers to the number of received messages.
- “Dropped” refers to the number of packets lost due to interference.
- “D/Total” is the number of dropped packets in relation to the total number of packets received, and dropped, in the system. This is a measure of the interference between the communicating vehicles.
- “D/Vehicle” is the number of dropped packets pr vehicle. This gives an average of how many packets each vehicle has been unable to receive.
- “D/TX” is the number of dropped packets pr sent packet. In a wireless system, each transmission has potentially several receivers, and it is the receiver that drops the packet.
- “D/RX” is the number of dropped packets pr successfully received packet. This gives an average of packet loss in the system.

[4] 100m < Distance between vehicles > 250m
[5] In 10^{-3} seconds (ms)

7. Discussion

The results in Table 1 are more clearly illustrated with the help of some graphs.
Graph 1 compares how the "Single Ripple" algorithm and the "pure flooding" algorithm performed in terms of how many packets were sent, received and dropped. The shape of the curves belonging to "pure flooding" is as expected, while "sent" and "received" from "Single Ripple" has a somewhat unexpected shape.



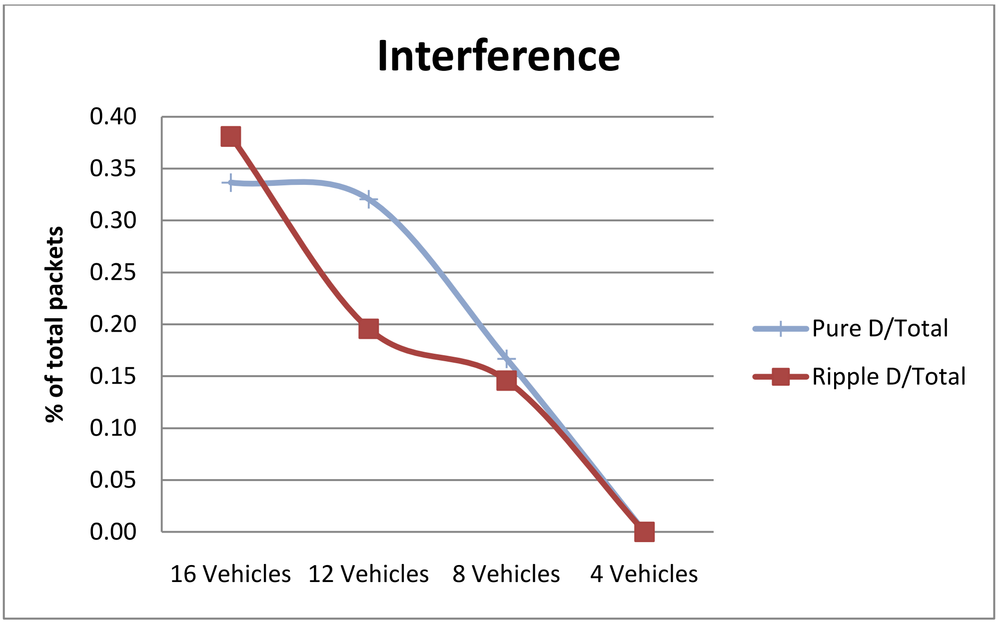
Graph 1 Complete presentation of sent, received and dropped packets

It is evident that the number of sent and received messages peaks at "8 Vehicles" when using the "Single Ripple" algorithm. The fact that the number of received messages drops with increasing numbers of vehicles is due to increased interference in the system. This limits the throughput in the system. In "pure flooding" the increased interference is masked by the sheer number of packets being transmitted.

As seen by Graph 1, "Single Ripple" reduced the number of messages being received by 91% compared to "pure flooding" (16 vehicles). Furthermore, "Single Ripple" sends 89% fewer packets than "pure flooding" with 16 vehicles. This means that the load on the scarce resources in the system is decreased.
"Pure flooding" involves more vehicles than is strictly necessary to get the requested information, but offers alternative routes of communication, making up for some of the lost packets. "Single Ripple" uses the resources in a small number of vehicles, and no resources in the others; "pure flooding" uses the same amount of resources in all the vehicles.
Based on the values from Table 1 it is evident that the vehicles involved in "Single Ripple" spend far less time processing messages; the time is reduced by 91% compared to "pure flooding" (16 vehicles).

As can be seen in Graph 2, "Single Ripple" reduces the amount of interference at 12 vehicles, but it increases again at 16 vehicles compared with "pure flooding". "Pure flooding" has an unexpected slow rise from 12 to 16 vehicles, but this can be attributed to the fact that most of the new vehicles from 12 to 16 are placed at the edge of the simulation area. The position of the new vehicles means that they contribute to the total number of messages without increasing the interference to the same effect, and thus "pure flooding" seems less prone to interference within the area than "Single Ripple" when more vehicles are involved.

Each sent and received message in “Single Ripple” contributes more to the result than does each message in “pure flooding” since the information is specific to the conditions we want to explore. This fact also makes each dropped packet more valuable. The communication protocol used in the system is UDP, which was chosen because TCP introduces an extra load. No retransmission of lost or damaged packets is a problem when using “Single Ripple”, as the algorithm aims at involving as few vehicles as possible. The possibilities of receiving the information via alternate routes are slim. In “pure flooding” there is more redundancy, and therefore it is more likely that the information will be received through several different routes. The presence of several identical messages increases the potential for the receivers to actually receive the message.



Graph 2 Dropped packets out of the total number of packets

At least two solutions to the problem of lost messages in “Single Ripple” can be suggested:

- Let the vehicles update each other at regular intervals. This would, over time, cover the holes left by lost messages.
- Use the fact that the vehicles move. After the first run, let some time pass before the same discovery is initiated again. After 1 second, with an average of 30 km/h, the vehicles will have moved 8 meters, and after 10 seconds they will have moved 80 meters. This might be enough to change the pattern of packet loss, and thereby gain new information.

“Pure flooding” is known to be problematic when it comes to efficiency, but it is the only alternative algorithm that was suitable for comparison against “Single Ripple” when it comes to discovering the border. The simulations have been carried out on a small selection of vehicles to show the principles behind the algorithm. With larger numbers of vehicles the amount of interference would increase, especially with “pure flooding” when the entire network is involved, and the damaging effect of lost messages in “Single Ripple” would be reduced.

Security related to “Single Ripple” is not considered, as this is more related to how the messages are processed before and after transmission. The reliability of “Single Ripple” is dependent upon the density of the vehicles around the conditions in question, and will increase with higher densities, as there will be more retransmissions and more details available.

8. Conclusions

This paper presents new vehicle-to-vehicle communication protocol capable of discovering areas with a specific traffic condition – e.g. congestion, slippery area, rain etc. The protocol is highly efficient – it shows appr. 10 times better results than “pure flooding” protocol. It is infrastructure-less, but this is not a restriction – it can make use of any additional nodes alongside the road.
The protocol is intended to be used in all collaborative schemes for cooperative vehicle information generation and traffic information distribution.

9. References

1. Cooperative Vehicle Infrastructure Systems – CVIS, EU FP6 Project Reference: IST-2004-027293, Contract Type: Integrated Project (IP), Project Cost: €41.155.203, EC project funding: €21.905.795.
2. David R. Choffnes , Fabián E. Bustamante, "An integrated mobility and traffic model for vehicular wireless networks", Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks, September 02-02, 2005, Cologne, Germany
3. Gökhan Korkmaz, Eylem Ekici, Füsun Özgüner, Ümit Özgüner "Urban multi-hop broadcast protocol for inter-vehicle communication systems", VANET '04 Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks ©2004 table of contents ISBN:1-58113-922-5 doi>10.1145/1023875.1023887
4. JA Fax, RM Murray, "Information flow and cooperative control of vehicle formations - IEEE Transactions on Automatic Control, 2004
5. Joon-Sang Park , Uichin Lee , Soon Young Oh , Mario Gerla , Desmond Siumen Lun , Won Woo Ro , Joonseok Park, "Delay Analysis of Car-to-Car Reliable Data Delivery Strategies Based on Data Mulling with Network Coding", IEICE - Transactions on Information and Systems, v.E91-D n.10, p.2524-2527, October 2008
6. Molisch, A.; Turfvesson, F.; Karedal, J.; Mecklenbrauker, C., "Propagation Aspects of Vehicle-to-Vehicle Communications - An Overview", *IEEE Radio & Wireless Symposium*, January 2009
7. Thomas, M.; Peytchev E.; Al-Dabass D.; "Auto-sensing and distribution of traffic information in vehicular ad hoc networks", International Journal of Simulation, January 2004, PP 59-63, Volume: 5(3), ISSN: 1473-804X
8. Wai Chen Shengwei Cai , **"**Ad hoc peer-to-peer network architecture for vehicle safety communications", April 2005, Volume: 43, Issue: 4, page(s): 100- 107, ISSN: 0163-6804, INSPEC Accession Number: 8375775, Digital Object Identifier: 10.1109/MCOM.2005.1421912, Current Version Published: 2005-04-18
9. Xue Yang, Jie Liu, Feng Zhao, and Nitin Vaidya, "A Vehicle-to-Vehicle Communication Protocol for Improving Road Safety," The 1st International Conference on Mobile and Ubiquitous Systems: Networking and Services (Mobiquitous 2004), Boston, MA, Aug. 22-26, 2004.
10. Yiannos Mylonas , Marios Lestas , Andreas Pitsillides, Speed adaptive probabilistic flooding in cooperative emergency warning, Proceedings of the 4th Annual International Conference on Wireless Internet, November 17-19, 2008, Maui, Hawaii