I-CROSS: Intersection Crossing Warning Application Based on V2V Communications

Unai Hernandez-Jayo
DeustoTech-Mobility Lab
Deusto Institute of Technology
Av Universidades 24, 48007 Bilbao Spain
Email: unai.hernandez@deusto.es

Idoia De-la-Iglesia
DeustoTech-Mobility Lab
Deusto Institute of Technology
Email: idoia.delaiglesia@deusto.es

Iñigo Lacoume
DeustoTech-Mobility Lab
Deusto Institute of Technology
Email: inlacou@gmail.com

***Abstract*—I-CROSS is an intelligent system designed to help drivers to decide the time slot in which an intersection can be crossed without collisions. This service has been developed as a desktop application that can be executed in Android devices. The set of algorithms that determine the crossing order are fed with information about surrounding vehicles: actual and further GPS position, speed and an identification number. The interchange of this information relays in a IEEE 802.11p communications architecture, that is, in a VANET network created ad-hoc among the vehicles are close to the intersection. The reliability of the application depends on the probability that the vehicles would be able to share this information, so before testing I-CROSS, the communication link must be checked in different scenarios. In this paper the results of these tests are shown and a description of the I-CROSS is also included.**

I. Introduction

A large amount of vehicle accidents occurs at road crossings or intersections. According to [Staubach, 2009] vehicle collisions at intersections are between 25% and 45% of all of them. Since intersections represent a very small portion of the roadway, so this number of accidents is considered a disproportionate amount. Due to that, intersection safety remains a challenge both for Advanced Driver Assistance Systems (ADAS) and autonomous driving [Spek et al., 2006].

According to statistical studies, the 90% of them are due to driver errors [Lefevre et al., 2012], as perception failures (e.g. inattention), misunderstanding (e.g. misjudging the intentions of another driver), and wrong decision (e.g. incorrect maneuver).

In this context, Intelligent Transportation Systems (ITS) can provide information and robotic techniques to increase the safety and driving efficiency. In the automotive industry, sensors are mainly used to give information to the driver, or to advise him about the presence of a dangerous situation [Wang et al., 2010]. In some cases, these automatic systems are connected to a computer that performs some guiding actions, attempting to minimize injuries and prevent collisions [Bishop, 2000].

ITS can play a key role in order to avoid these hazardous situations, using two main technical structures: first, a communication architecture is required in order to allow vehicles to exchange and share information about their position and driving parameters (speed, acceleration, direction). Secondly, a set of intelligent management algorithms defined to minimize accidents advising the driver about the presence of a dangerous area, or acting over the vehicle actuators [Papadimitratos et al., 2009].

In this paper we present I-CROSS, an intersection collision warning and cooperative awareness application that, using the facilities provided by a communications platform deployed using the IEEE 802.11p wireless standard, is able to advice the driver about the best option to cross the intersection, depending on the positions and directions of surrounding vehicles.

In section II the vehicle onboard requiered architecture to deploy ICROSS is described. Then, the communications infraestructure and the algorithms defined in ICROSS are analyzed. In section III the reliability of ICROSS is checked and finally conclusions are future work are presented.

II. System architecture

I-CROSS is based on an IEEE 802.11p communications architecture, which allows vehicles to exchange information, being this technology the best one that satisfies the requirements of a dynamic and changing scenario as it is created by moving nodes [Wu et al., 2013].

The topology of these networks can be centralized in an element located in the infrastructure (V2I and I2V links) or it can be distributed (mainly based in V2V link), making possible that the vehicles create ad-hoc connections among them.

I-CROSS has been designed to be deployed without the requirement of a central node. We have taken this decision for the following reasons:

- The messages shared among the vehicles close to the intersections are considered safety-critical, so in order to reduce the time delay among nodes and to satisfy the requirements of ETSI TC on ITS [ETSI, 9 06], I-CROSS is based on vehicle-to-vehicle links.
- To allow a fast deployment of I-CROSS and make it cheaper, we consider that a coverage of the road means RSU is not affordable so V2V is the best solution to provide cooperative applications at these particularly dangerous road sections.

Although is it out of the scope of this work, the algorithms of I-CROSS are independent of the communications platform thus just in case VANETs networks would not be available,

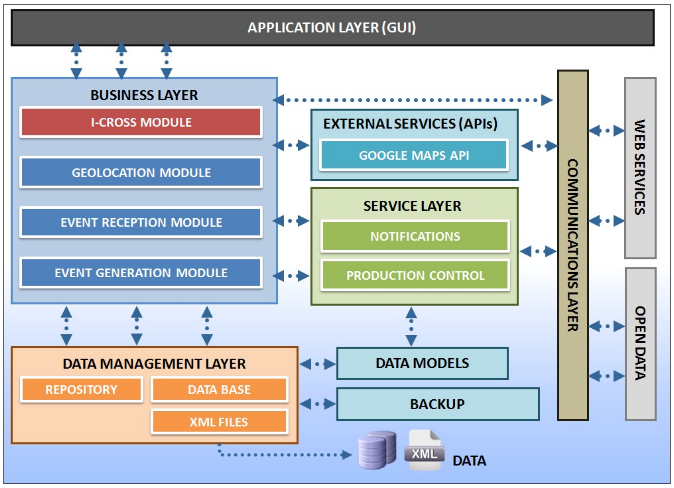

Fig. 1. I-CROSS software architecture

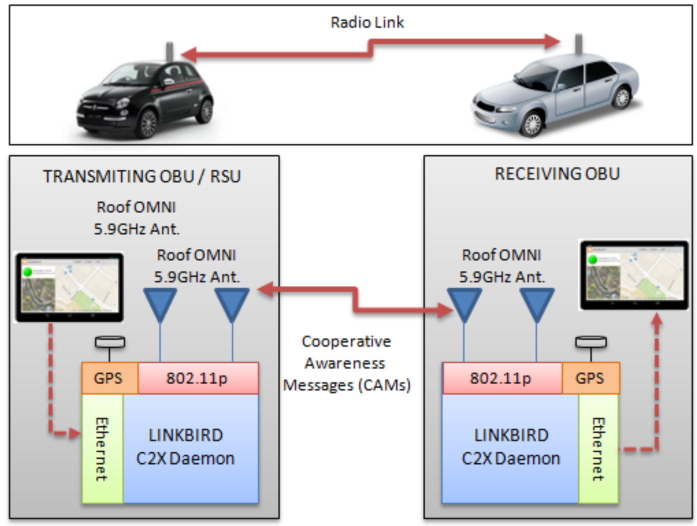

Fig. 2. Hardware Setup

common mobile networks could be used to exchange message among vehicles [Yaacoub and Zorba, 2013] , but in this case the communications delay restrictions could not be satisfied [Badawy et al., 2010].

As it is described at section II-A each vehicle is provided with a communications gateway (NEC LinkBird-Mx) and a nomad device as a tablet where I-CROSS application runs. To make possible the deployment of further applications and services, the system's software architecture has been designed using a layer model as it is shown at Figure 1. In this way, all the tasks that have to be controlled by the system are independent and can be updated in future versions. The main layers of the defined architecture are:

- Application layer is the user graphical interface. As it is described in section II-C, in order to not distrurb the driver, the interface is easy to use. It provides visual and audible notifications.
- Business layer which contents all the modules that can be provided in a vehicular cooperative scenario as slow vehicle ahead, road works alert, dynamic routing, etc. Common modules as event reception and generation have been defined because these are used by the other modules gathered in this layer.
- Data management layer supports the repository and data base of the system. This is a small layer designed to be deployed in nomad gadgets, but if powerful devices are available, this layer could support more data and complex data base. Its task is to record and storage relevant events that occur during the driving.
- The service layer manages the data flow between the communication layer and the business layer. It has been created in order to be easily updated in case the communication gateway changes or other data providers are supported by the system.
- Communications layer controls the data traffic from all the sources of the system: IEEE 802.11p communication gateway, available open data repositories (in this work in progress http://opendata.euskadi.net/ is used as traffic data provider to update the system with events on the road) or web services, as for example a Traffic Management Center that control an specific road area. This means that although I-CROSS is based on V2V links, the system also can support V2I and I2V links for others applications.

In this architecture I-CROSS module is the one that determines the crossing order in the intersections, thanks to the information received from surrounding vehicles and provided by the communications layer through the notification service.

A. *Communications Setup*

As we explained before, in order to develop I-CROSS we implemented vehicular communications using IEEE 802.11p which is the technology designed for vehicular networks. I-CROSS is a cooperative vehicular application because it allows the information exchange among vehicles with the ultimate aim of achieving benefits for many areas of traffic management and road safety. But to deploy this cooperative application some hardware must be installed on the vehicle to implement the wireless communication among vehicles. The main element is a vehicular communication router which implements IEEE 802.11p. We selected LinkBird-MX which embedded Linux machines based on a 64 bits MIPS processor working at 266MHz. Besides an IEEE 802.11p interface, these modules are equipped with an Ethernet connector that is used to communicate with the Application Unit (the one that runs the applications in a regular PC or a Tablet), a GPS interface and other interfaces as CAN or RS-232. We used DGPS to ensure that the vehicle's position has the less error possible. Figure 2 shows the hardware setup used in the tests that have been carried out in the described scenarios.

Although LinkBird allows to select two channel bandwidth, in these tests 10MHz bandwidth has been selected instead of the 20 MHz one usually used by 802.11a devices, in order to minimize multipath delay spread and Doppler effect that appears in mobility and highway scenarios. Moreover, in order to maintain sufficient reliability of the data transfer in

TABLE I. IEEE 802.11P HARDWARE SETUP

LinkBird-MX		Antenna	
Parameter	Values	Parameter	Values
Frequency	5725-5925 MHz	Model	ECO6-5500
Bandwidth	10MHz	Frequency	5.0-6.0 GHz
Tx Power	21dBm	Gain	6dBi
Bitrates	3Mbps	Radiation	Omni-directional

a 1-hop scenario, the lowest bit rate has been used, that is 3Mbps (bit rates from 3 to 27Mbps are available at IEEE 802.11p standard), so also the lowest coding rate (1/2) with BPSK modulation has been used to transmit data packets. Along with the communication modules, two antennas whose characteristics fit well with vehicular applications are provided. One antenna is tuned at 5.890GHz, which is used to send control messages and high priority messages, and the other one is tuned at 5.9GHz, which is used to send service messages. Technical characteristics of hardware setup are shown at Table I

B. Collision warning algorithm

As in other related works [Bento et al., 2012], the behaviour of I-CROSS has also been tested in a simulator. The purpose of the simulator which can be seen in Figure 3 is to test the algorithms designed for the deployment of I-CROSS and to determine the crossing order in an intersection.

As shown in Figure 3, the simulator is rather simple. It allows the user to create a custom map and then launch a test. Each vehicle type has it's own size, the ambulance has a different coloured rectangle inside its shape and pedestrians are drawn as a stick with a slightly gray head. The main color of each vehicle or pedestrian is determined by the algorithm. If an alert has been thrown the color will be turn into red (darker), and turn into green (clearer) otherwise. The zoom of an intersection at the simulator on Figure 3 can help to understand the algorithm.

Once the testing phase finished, the algorithms were easily ported to nomad devices with Android Operating System. The algorithm's operating rules allow it to:

- Include high priority vehicles (ambulance, police, fire-fighter etc.).

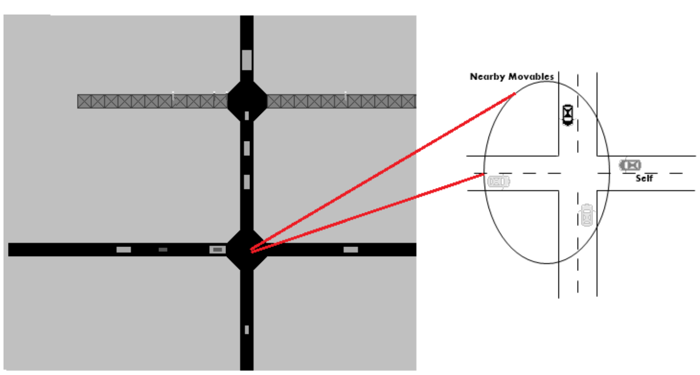

Fig. 3. Simulator Screenshot

```
Data: self, nearbyMovables
Result: alert, noAlert
initialization;
foreach m in nearbyMovables do
    if route_self intersects route_m then
        if self has priority then
            calculate intersection reach and leave
            time for self and m
            calculate secure time frame
        end
    else
        if self is firstInRoute then
            if place not available after the
            intersection then
                return alert
            end
        end
    end
end
return noAlert
```

Algorithm 1: Base algorithm

- Pedestrians (which will allow in the future to include a scenario where pedestrians could communicate their position or proximity to a pedestrian crossing).
- Alert users of possible imminent accidents.
- Adjust the passing order to avoid jams.
- Consider motorcycles, trucks and regular vehicles with certain lengths.

The algorithm used by I-CROSS uses data received from each vehicle to calculate the arriving and leaving time in the intersection. The type of each vehicle is also significant for the system as it tries to give passing priority to emergency vehicles. A simplification of the main algorithm is represented in the Algorithm 1.

The first *'if'* determines if two vehicles' routes cross each other, because if the vehicles do not intersect there is no calculation to be done. In case the first *if* is passed, the algorithm checks whether the other vehicle has priority. This happens in the following cases:

- Self vehicle is a conventional vehicle and...
  - The other vehicle approaches from the right.
  - The other vehicle is an emergency vehicle.
- Self vehicle is an emergency vehicle and the other vehicle is an emergency vehicle which approaches from the right.

If it passes the second *'if'*, then calculates the approximation and leaving time to the intersection of each vehicle and passes it to another function represented in Algorithm 2.

This Algorithm 2 determines whether the user can enter and leave the intersection safely. Compares the arriving (*rt*) and leaving (*lt*) time of both vehicles and alerts if a collision is about to happen. The margin time (*mt*) attribute defines a security time frame. For example, if the arriving time of the first vehicle is shorter than the leaving time of the

Data: rt_1, lt_1, rt_2, lt_2
Result: $alert, noAlert$
if $rt_1 > 0$ **then**
  if $rt_2 > 0$ **then**
    if $rt_2 < lt_1$ **then**
      return $alert$
    else if $|rt_2 - lt_1| < mt$ **then**
      return $alert$
    end
  else if $|rt_2 - rt_1| < mt$ **then**
    return $alert$
  end
  if $lt_2 > 0$ ***AND*** $(rt_2 < 0$ ***OR*** rt_2 ***is*** $null)$ **then**
    if $rt_1 < lt_2$ **then**
      return $alert$
    else if $|rt_1 - lt_2| < mt$ **then**
      return $alert$
    end
  end
end
return $noAlert$

Algorithm 2: Calculate secure time frame Algorithm

second vehicle means that the first vehicle is going to enter the intersection while the second vehicle is into it, and the algorithm throws an *alert*.

The Algorithm 3 calculates if the vehicle which is passed as a parameter has other vehicles in front of it before the intersection. This algorithm is used by the main algorithm when it seems the vehicle could cross before checking whether there is room for the vehicle after the intersection.

C. User Interface

The application's interface aims to be rather simple. It represents two maps and a *virtual traffic light*, each one in it's own rectangle. The small map shows the general route of the user so that it can observe the state of his journey in a glance. The biggest map shows information about the surroundings of the user's vehicle. It provides information of nearby vehicles, such as vehicle type and whether he should be concerned about them or not. If there's any car, the user should be concerned about, the *virtual traffic light* will turn red, an alert sound will be played and the vehicle will turn from green to red in the big map. This traffic light will also display some information to the user like his speed or the distance until the next crossing. In this way, the application does not require interaction from the user, apart from initial configurations.

Data: $movable, intersectionPoint, movablesList$
Result: $true, false$
foreach m ***in*** $movablesList$ **do**
  if m ***not equals*** $movable$ **then**
    if $posList_m$ ***contains*** $intersectionPoint$ ***AND*** $posList_movable$ ***contains*** $position_m$ **then**
      if $posList_m.index(position_m) + m.size < posList_m.index(intersectionPoint)$ **then**
        if $(posList_m.index(intersectionPoint) - posList_m.index(position_{movable})) > (posList_{movable}.index(intersectionPoint) - posList_{movable}.index(position_m))$ **then**
          return $false$
        end
      end
    end
  end
end
return $true$

Algorithm 3: firstInRoute Algorithm

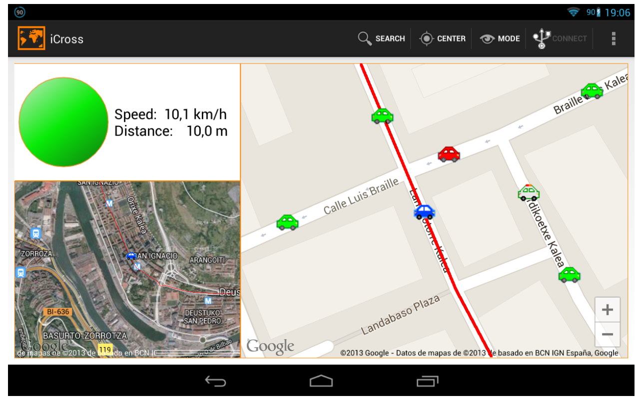

Fig. 4. Onboard user interface

III. APPLICATION EVALUATION

As we mentioned before, the validation of application reliability is necessary to deploy cooperative applications. This reliability depends on the reached goodput of the IEEE 802.11p radio link. Therefore, in order to demonstrate the application reliability the radio link was characterized in two different scenarios: a crossroads where there is line-of-sight (LOS) between vehicles and a crossroads where there is no-line-of-sight (NLOS) between vehicles because of the buildings. The selected scenarios are shown at Figure 5.

A Java application which sends packets with 100 bytes of payload was developed. The sending frequency is also configurable, from 1Hz to 10Hz. This application also measures the delay of a packet from the source to the destination. Each packet is formed by a header, which contents vehicle identification, GPS position and speed, and a payload with the next 10 GPS position of the driver's route. These geographical

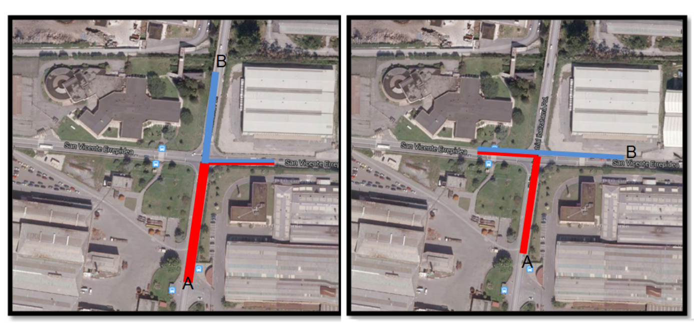

Fig. 5. Tested Scenarios

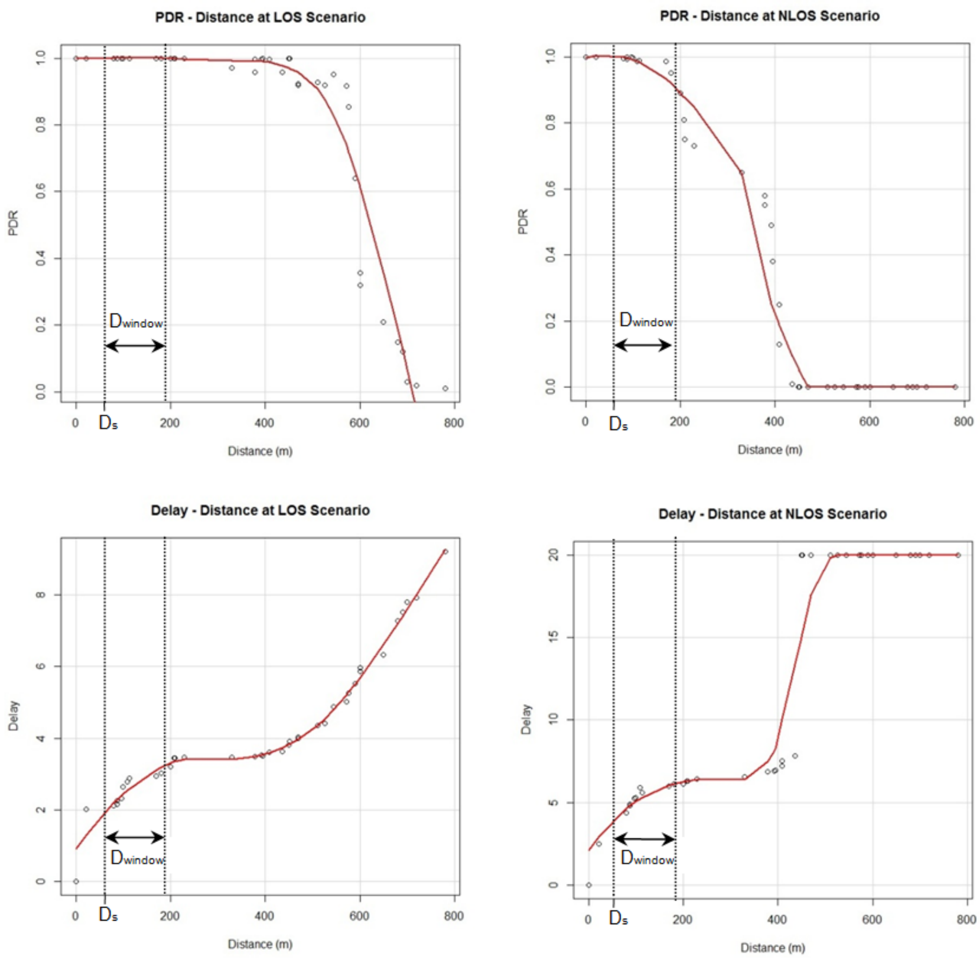

Fig. 6. PDR measurements at diferent distances and LOS and NLOS scenarios

points are used by the I-CROSS module to determine if two or more vehicle will crash in the intersection. An essential characteristic to validate the IEEE 802.11p communications is the Packet Delivery Rate (PDR) at a target distance, since for these cooperative applications make sense messages must be received before the vehicle arrives at the event being reporting (an accident, a traffic jam, etc.). In I-CROSS the messages of other vehicles approaching the same crossroad must be received before a target distance (Ds) far to the collision point.

Ds is the distance needed by a driver to stop the vehicle. To calculate this distance is necessary to take into account the driver's reaction time and the braking distance. Using equation 1, where V is the velocity expressed in Km/h, f represents the friction coefficient and i the slope of the road in %, we calculate the braking distance. Considering that the vehicle travels at 50km/h it requires at least 24m to stop in a flat road. We consider 50 km/h because this is the maximum speed allowed in urban environments in Spain. According to [Triggs et al., 1982], drive's reaction time can be from 1.26s to 3s. If we consider an average value of 2.5s, in this time car travels 34m, so in total driver needs 57m to stop the car.

$$D_s = \frac{V^2}{254(f+i)} \tag{1}$$

In Figure 6 the PDR and delay measurements obtained at different distances in the two evaluated scenarios are shown. These results are in a scatter plot to show the relationship between the PDR and delay with the distance between vehicles.

To check the correlation between PDR and distance and delay and distance the Pearson correlation was obtained with the equation 2, where σ_{XY} is de covariance, σ_X is the standard desviation of X variable and σ_Y is the standard desviation on Y variable. We obtained that the Pearson coefficient for the correlation between PDR and distance is 0,7754 and 0,9137 for the correction between delay and distance. Due to these results we can conclude that the parameters are correlated because they are greater than 0,7 [2]. The values obtained make some sense because the delay is directly related with the distance that the packet has to travel from a source to a destination, and the probability of receiving a packet is greater when the vehicles are closer. In addition, the correlation between delay and distance is greater than the correlation between PDR and distance, this is owing to the PDR is always 1 until the vehicles distance is greater than a value.

$$\rho_{X,Y} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y} \tag{2}$$

As the PDR is not always 1, the Cooperative Awareness

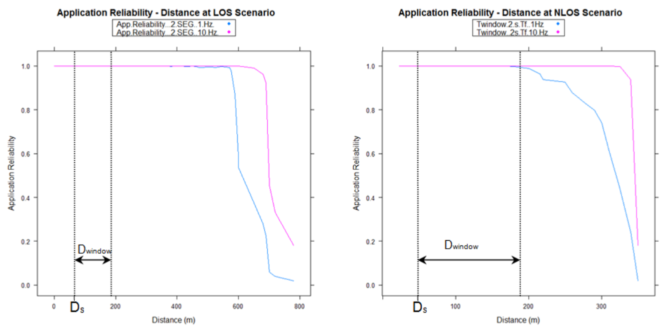

Fig. 7. Application reliability at LOS and NLOS scenario

Message (CAM) should not be exchanged between vehicles only once, so it is send repeated times. The easiest way would be to send it repeatedly until reach the crossroad, although this is not viable since they could overload the radio link. Then we have to define the ideal number of times, N_T, that the message must be resent for the application to be reliable. N_T depends on two different values, which are the message sending frequency, Tf, and the Twindow which is a concept proposed to evaluate the applications reliability. For this type of advisor application the Twindow is 2.0 seconds [Bai and Krishnan, 2006].

$$N_T = Tf * Twindow \tag{3}$$

Twindow is related with Dwindow, which is the distance that the vehicle travels during the time defined by Twindow. At the same time, Dwindow is related with the application reliability because it needs to be defined to quantify the probability of successfully communicating the two vehicles before the target distance Ds.

$$Dwindow = Twindow * Vvehicle \tag{4}$$

The application reliability (p_{app}) is defined as the probability of receiving at least one packet before Ds in a given time Twindow.

$$p_{app} = 1 - \prod_{k=1}^{k=N_T} (1 - p_i) \tag{5}$$

At the Figure 7 the application reliability results of the real tests done are shown. It is shown that at 57 meters far from the collision point, PDR value is '1' in any case (LOS or NLOS scenario), so we can consider the application is reliable.

IV. Conclusion and Future Work

According to the results obtained in the previous section, we can conclude that a communications architecture based on the NEC-Linkbirds can be used to deploy cooperative awareness applications as I-CROSS, in both LOS and NLOS scenarios.

It must also be considered that the application reliability depends on the CAM transmission frequency. As it is has been evaluated, the p_{app} changes its probability of success if the transmission frecuency changes from 10Hz to 1Hz. In a NLOS scenario where the transmission frequency is setup to 10Hz, the application reliability is 100%, but in a scenario with a high density of vehicles, the communications link will be overload due to the huge amount of sending messages. In this situation, techniques of MAC must be deployed in order to allow every vehicle to exchange information with the maximun application reliability.

V. Acknowledgment

The authors would like to thank the EU Intelligent Cooperative Sensing for Improved traffic efficiency (ICSI) project (FP7-ICT-2011-8) for its support in the development of this work.

References

[Badawy et al., 2010] Badawy, G., Misic, J., Todd, T., and Zhao, D. (2010). Performance modeling of safety message delivery in vehicular ad hoc networks. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2010 IEEE 6th International Conference on*, pages 188–195.

[Bai and Krishnan, 2006] Bai, F. and Krishnan, H. (2006). Reliability analysis of dsrc wireless communication for vehicle safety applications. In *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, pages 355–362.

[Bento et al., 2012] Bento, L., Parafita, R., and Nunes, U. (2012). Intelligent traffic management at intersections supported by v2v and v2i communications. In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pages 1495–1502.

[Bishop, 2000] Bishop, R. (2000). A survey of intelligent vehicle applications worldwide. In *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*, pages 25–30.

[ETSI, 9 06] ETSI (2009-06). *Intelligent Transport Systems (ITS) and Vehicular communications and Basic set of applications and Definitions*. ETSI TR 102 638 V1.1.1.

[Lefevre et al., 2012] Lefevre, S., Laugier, C., and Ibanez-Guzman, J. (2012). Evaluating risk at road intersections by detecting conflicting intentions. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4841–4846.

[Papadimitratos et al., 2009] Papadimitratos, P., La Fortelle, A., Evenssen, K., Brignolo, R., and Cosenza, S. (2009). Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation. *Communications Magazine, IEEE*, 47(11):84–95.

[Spek et al., 2006] Spek, A., Wieringa, P., and Janssen, W. (2006). Intersection approach speed and accident probability. *Transportation Research Part F: Traffic Psychology and Behaviour*, 9(2):155 – 171.

[Staubach, 2009] Staubach, M. (2009). Factors correlated with traffic accidents as a basis for evaluating advanced driver assistance systems. *Accident Analysis & Prevention*, 41(5):1025 – 1033.

[Triggs et al., 1982] Triggs, T., Harris, W., of Psychology, M. U. D., Group, M. U. H. F., and of Road Safety, A. O. (1982). *Reaction Time of Drivers to Road Stimuli*. Human factors report. Human Factors Group, Department of Psychology, Monash University.

[Wang et al., 2010] Wang, J., Xu, W., and Gong, Y. (2010). Real-time driving danger-level prediction. *Engineering Applications of Artificial Intelligence*, 23(8):1247 – 1254.

[Wu et al., 2013] Wu, X., Subramanian, S., Guha, R., White, R., Li, J., Lu, K., Bucceri, A., and Zhang, T. (2013). Vehicular communications using dsrc: Challenges, enhancements, and evolution. *Selected Areas in Communications, IEEE Journal on*, 31(9):399–408.

[Yaacoub and Zorba, 2013] Yaacoub, E. and Zorba, N. (2013). Enhanced connectivity in vehicular ad-hoc networks via v2v communications. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, pages 1654–1659.